

NASA/TM-2000-208641/VER1.0/VOL#

## **ICESat (GLAS) Science Processing Software Document Series**

### **Volume # GSAS User's Guide Version 1.0**

**Jeffrey Lee/Raytheon ITSS  
Observational Science Branch  
Laboratory for Hydrospheric Processes  
NASA/GSFC Wallops Flight Facility  
Wallops Island, Virginia 23337**

**November 2000**

#### **ICESat Contacts:**

**H. Jay Zwally, ICESat Project Scientist  
NASA Goddard Space Flight Center  
Greenbelt, Maryland 20771**

**Bob E. Schutz, GLAS Science Team Leader  
University of Texas Center for Space Research  
Austin, Texas 78759-5321**

**David W. Hancock III, Science Software Development Leader  
NASA/GSFC Wallops Flight Facility  
Wallops Island, Virginia 23337**





## **Foreword**

This document contains the GLAS Science Algorithm Software (GSAS) User's Guide. This document is developed under the structure of the NASA STD-2100-91, a NASA standard defining a four-volume set of documents to cover an entire software life cycle. Under this standard a section of any volume may, if necessary, be rolled out to its own separate document. This document is a roll-out of the user guide within the Product Specification Volume.

The GEOSCIENCE LASER ALTIMETER SYSTEM (GLAS) is a part of the EOS program. This laser altimetry mission will be carried on the spacecraft designated EOS ICESat (Ice, Cloud and Land Elevation Satellite). The GLAS laser is a frequency-doubled, cavity-pumped, solid state Nd:YAG laser.

This document was prepared by the Observational Science Branch at NASA GSFC/WFF, Wallops Island, VA, in support of B. E. Schutz, GLAS Science Team Leader for the GLAS Investigation. This work was performed under the direction of David W. Hancock, III, who may be contacted at (757) 824-1238, hancock@osb.wff.nasa.gov (e-mail), or (757) 824-1036 (FAX).

The following GLAS Team members contributed to the creation of this document:

Raytheon/Kristine Barbieri

Raytheon/Suneel Bhardwaj

Raytheon/Anita Brenner

972/David W. Hancock, III

Raytheon/Peggy Jester

Raytheon/Gladstone Marcus

Raytheon/Carol Purdy

Raytheon/Lee Anne Roberts



# Table of Contents

Foreword .....	iii
Table of Contents .....	v
List of Figures .....	vii
List of Tables .....	ix
<b>Section 1        Introduction</b>	
1.1     Identification of Document .....	1-1
1.2     Scope of Document .....	1-1
1.3     Purpose and Objectives of Document .....	1-1
1.4     Document Organization .....	1-1
1.5     Document Change History .....	1-2
<b>Section 2        Related Documentation</b>	
2.1     Parent Documents .....	2-1
2.2     Applicable Documents .....	2-1
2.3     Information Documents .....	2-1
<b>Section 3        Overview</b>	
3.1     Purpose .....	3-1
3.2     Environment .....	3-1
3.3     Functions .....	3-1
3.4     Restrictions and Limitations .....	3-2
<b>Section 4        Installation and Initialization</b>	
4.1     Unpack the Tarfile .....	4-1
4.2     Compilation .....	4-1
<b>Section 5        GLAS_Exec</b>	
5.1     Startup and Termination .....	5-1
5.2     Functions and Their Operation .....	5-2
5.3     Error and Warning Messages .....	5-6
5.4     Recovery Steps .....	5-7
<b>Section 6        Product Readers/Writers</b>	
6.1     Startup and Termination .....	6-1
6.2     Functions and Their Operation .....	6-1
6.3     Fault Recovery Procedures .....	6-5
6.4     Emergency Procedures .....	6-6
6.5     Diagnostic Procedures .....	6-6
<b>Appendix A      GLAS File Information</b>	
A.1     GLAS Production File Naming Convention .....	A-1
A.2     EDOS L0 File Naming Convention .....	A-2

A.3	GLAS File Type List.....	A-3
A.4	Product Granule Sizes .....	A-4

**Appendix B ANC07 Format, Files and Contents**

B.1	ANC07 Format .....	B-1
B.2	ANC07 Files .....	B-2
B.3	ANC07 Contents.....	B-2

**Appendix C GLAS\_Exec Control File Format and Elements**

C.1	Control File Format .....	C-1
C.2	Start and Stop Time Designation .....	C-5
C.3	Required Control File Elements .....	C-5

**Appendix D Sample Control Files**

D.1	Sample L1A Control File .....	D-1
D.2	Sample Waveform Control File .....	D-5
D.3	Sample Atmosphere Control File.....	D-10
D.4	Sample Elevation Control File.....	D-13

**Appendix E GSAS ANC06 File Contents**

E.1	ANC06 Overview .....	E-1
E.2	ANC06 Example and Description .....	E-1

**Appendix F GSAS Error and Status Codes**

F.1	Invalid Values.....	F-1
F.2	Error/Status Format.....	F-1
F.3	Error/Status Usage.....	F-2
F.4	GSAS Error Codes .....	F-3
F.5	HP Runtime Error Codes.....	F-9

Abbreviations & Acronyms..... AB-1

Glossary..... GL-1

## **List of Figures**

Figure 1-1	I-SIPS Software Top-Level Decomposition.....	1-2
Figure F-1	Error Ancillary File Format .....	F-1



## List of Tables

Table 4-1	GSAS Directory Structure .....	4-1
Table 6-1	Product Reader Output Files .....	6-2
Table 6-2	Product Writer Output Files .....	6-3
Table 6-3	Product Writer Output File Contents .....	6-4
Table 6-4	GLAS File Naming Keys .....	A-1
Table 6-5	EDOS L0 File Naming Keys .....	A-2
Table 6-6	GLAS File List .....	A-3
Table 6-7	Product Granule Sizes .....	A-4
Table B-1	ANC07 Granule Names and Contents .....	B-2
Table B-2	L1A Constants .....	B-3
Table B-3	Elevation Constants .....	B-9
Table B-4	Waveform Constants .....	B-9
Table B-5	Atmosphere Constants .....	B-12
Table B-6	Global Constants .....	B-14
Table C-1	Required Single-Instance Keywords .....	C-1
Table C-2	Optional Multiple-Instance Keywords .....	C-1
Table C-3	Waveform Control Flags .....	C-4
Table F-1	Invalid Values .....	F-1
Table F-2	Error String Format .....	F-1
Table F-3	Error Severity Codes .....	F-2
Table F-4	GLAS_Exec Error Sections .....	F-3



**Section 1**  
**Introduction**

## **1.1 Identification of Document**

This document is identified as the GLAS Science Algorithm Software (GSAS) User's Guide. The unique document identification number within the GLAS Ground Data System numbering scheme is \*TBD\*. Successive editions of this document will be uniquely identified by the cover and page date marks.

## **1.2 Scope of Document**

The GLAS I-SIPS Data Processing System, shown in Figure 1-1, provides data processing and mission support for the Geoscience Laser Altimeter System (GLAS). I-SIPS is composed of two major software components - the GLAS Science Algorithm Software (GSAS) and the Scheduling and Data Management System (SDMS). GSAS processes raw satellite data and creates EOS Level 1A/B and 2 data products. SDMS provides for scheduling of processing and the ingest, staging, archiving and cataloging of associated data files. This document is the User's Guide for GSAS and contains information specific to the Version 1 delivery of the software.

## **1.3 Purpose and Objectives of Document**

The purpose of this document is to provide the GSAS end users with instructions explaining how to operate the software effectively.

## **1.4 Document Organization**

This document's outline is assembled in a form similar to those presented in the NASA Software Engineering Program [Information Document 2.3a].

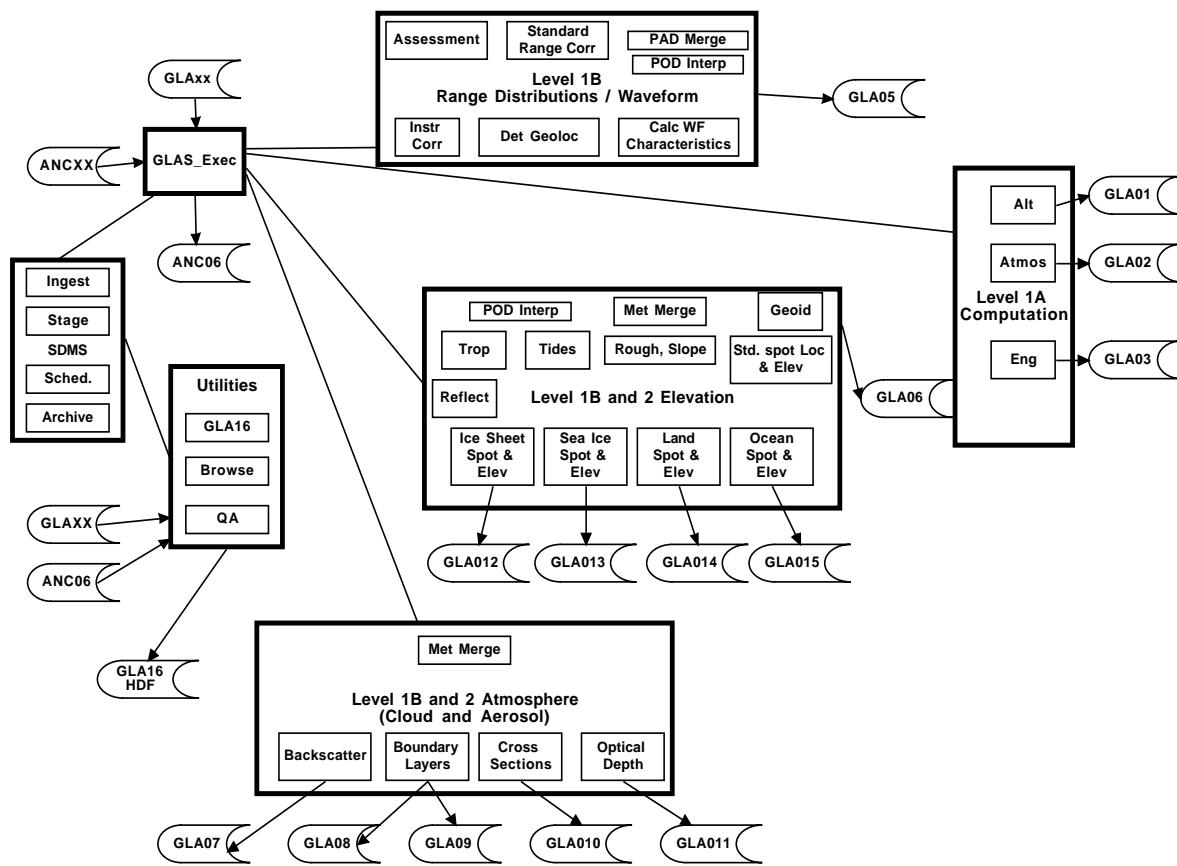


Figure 1-1 I-SIPS Software Top-Level Decomposition

## 1.5 Document Change History

Document Name: GLAS Science Algorithm Software Users' Guide		
Version Number	Date	Nature of Change
Version 0	August 1999	Original Version.
Version 1	November 2000	Revised for V1 software.

**Section 2**  
**Related Documentation**

## **2.1 Parent Documents**

Parent documents are those external, higher-level documents that contribute information to the scope and content of this document. The following GLAS documents are parent to this document.

- a) *GLAS Science Software Management Plan* (GLAS SSMP), Version 3.0, August 1998, NASA Goddard Space Flight Center, NASA/TM-1999-208641/VER3/VOL1.

The GLAS SSMP is the top-level Volume 1 (Management Plan Volume) document of the four volumes of NASA software engineering documentation [Applicable Reference 2.2c]. It dictates the creation and maintenance of the Product Specification Volume (Volume 2). This document is a roll out of the Product Specification Volume.

## **2.2 Applicable Documents**

- a) NASA Software Documentation Standard Software Engineering Program, NASA, July 29, 1991, NASA-STD-2100-91.
- b) GLAS Science Algorithm Software Detailed Design Document (GSAS DD), Version 1.0, September 2000, NASA Goddard Space Flight Center.
- c) GLAS Science Algorithm Software Version Description Document (GSAS VDD), Version 1.0, September 2000, NASA Goddard Space Flight Center.
- d) GLAS ISIPS Operational Procedures Manual, TBD.

## **2.3 Information Documents**

- a) GLAS Level 0 Instrument Data Product Specification, Version 2.2, March 17, 1998, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DPS-2610.
- b) GLAS Standard Data Products Specification - Level 1, Version 2.0, January, 1999, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DPS-2621.
- c) GLAS Standard Data Products Specification - Level 2, Version 2.0, January, 1999, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DPS-2641.
- d) GLAS Science Data Management Plan (GLAS SDMP), Version 4.0, June 1999, NASA Goddard Space Flight Center Wallops Flight Facility, GLAS-DMP-1200.



## Section 3

# Overview

### **3.1 Purpose**

GSAS generates the GLAS Standard Data Products and associated metadata describing the products and their quality. The software uses GLAS telemetry and ancillary data to produce the products using algorithms defined by the GLAS Instrument and Science Teams.

GSAS is delivered as a set of libraries and executables. The main processing software is GLAS\_Exec. It is accompanied by a set of utilities which perform various data transformations and computations.

Throughout this document, files are referenced as one of two types: GLA or ANC. GLA files are fixed-length, integer-binary format product files containing Level 0-2 GLAS science data. GLA files are both input and output to GSAS. ANC files are multi-format ancillary files supplied by the science team which are required for processing. A list of the GLA and ANC files is supplied in Appendix A. These files are detailed in the GLAS Data Management Plan and GLAS Standard Data Product Specifications Documents.

### **3.2 Environment**

GSAS software is developed for and delivered on the UNIX platform. This document assumes that the reader is familiar with UNIX operating system conventions. The software is currently supported only on the HP/UX 10.2 with Fortran 90 1.0 and HP/UX 11.0 with Fortran 2.3.

### **3.3 Functions**

The GSAS functions for V1 are:

- Read GLAS telemetry data and standard data products and ancillary files. Provide time-synchronization between product and ancillary files and between multiple products.
- Create all standard data products in an integer-binary format. These data products are grouped into the following categories:
  - Level-1A products. (GLA01-03)
  - Waveform products. (GLA05)
  - Atmosphere products. (GLA07-11)
  - Elevation products. (GLA06,GLA12-15)
- Perform selective processing based on input and output defined in a user-supplied control file.

- Create metadata which contains a full processing history.
- Report errors and messages in a standardized fashion with user-defined options available.
- Read changeable parameters from Science Team-supplied ancillary files.
- Convert product data into human-readable output.
- Create sample (but not scientifically accurate) test products.

### 3.4 Restrictions and Limitations

The V1 delivery of the GSAS has the following limitations:

- The software has the capability of processing many different scenarios. However, only tested scenarios are supported. These scenarios are:
  - One processing string to create all L1A products (GLA00 to GLA01-03).
  - One processing string that starts with a waveform analysis product (GLA05) input to produce all elevation products (GLA06, 12,13,14,15).
  - One processing string that starts with L1A atmosphere (GLA02) input and produces L2 atmosphere products (GLA07,08,09,10,11).
  - One processing string that starts with a waveform product (GLA05) input to produce a primary elevation product (GLA06).
- GSAS will **not overwrite** existing files. The software will halt with a fatal error unless old output files are removed before execution.
- Quality assurance output is not required for Version 1 and is implemented only in the Atmosphere and Waveform subsystems. The data in the resultant quality assurance files have not been fully tested.
- Only the integer-binary format of the GLAS standard data products will be generated. Headers will be added in the V2 delivery.
- It is assumed all input data are time-aligned with no missing data.
- No process sanity checking is delivered in this version. This will be added in the V2 delivery.
- GLA00 data will be in a test format. This format combines all APID files into a single file and provides header structures to ease data input. Support for the operational format of the GLA00 data will be delivered in a build with the GLAS L0 Processor (a separate utility).
- GLA00 data will not support a direct run of the software to create data from GLA00 through GLA15. Software support is implemented, but the team is not generating realistic GLA00 test data.
- Regional masks are not implemented. All data will be written to GLA12-15.

- GSAS is supported on HP/UX 10.2 with HP Fortran 90 1.0 and HP/UX 11.0 with HP Fortran 2.3 .
- In a production environment, the GSAS would be controlled by the SDMS. The SDMS would produce control files, stage data, and control execution of the GSAS binaries. This document, however, is limited to GSAS and thus will not describe procedures within the scope of SDMS.



Section 4

## Installation and Initialization

This section will detail installation instructions for GSAS. As GSAS will be delivered with HP/UX 11.0 binaries, recompilation is not required. Instructions, however, are provided in case the user wishes to recompile using debug or optimization options. Recompilation is not supported on different architectures since GSAS has only been tested under HP/UX 10.2/F90 1.0 and HP/UX 11.0/F90 2.3.

**Note:** Throughout this document, the text refers to directory names preceded by a “\$” and typed in all capital letters. This means that the directory path is local to the specific installation and not hard-coded within the GSAS. (This document uses standard UNIX ‘sh’ variable conventions to indicate such.) Other pathnames are specified relative to the local directories and only the relevance is important.

### 4.1 Unpack the Tarfile

Create a production directory for the GLAS\_Exec program. This directory is designated \$GLAS\_HOME.

```
mkdir $GLAS_HOME
```

Copy the delivered tar file into GLAS\_HOME and untar it. Designate \$DIST\_DIR as the source of the gsas\_v1.tar tarfile.

```
cd $GLAS_HOME  
cp $DIST_DIR/gsas_v1.tar.  
tar xf gsas_v1.tar
```

The untar process will create the directory structure described in Table 4-1.

**Table 4-1 GSAS Directory Structure**

Directory	Description
\$GLAS_HOME/bin	GSAS executables. Executables should always be linked from this directory
\$GLAS_HOME/lib	GSAS shared libraries. (The user should always set the SHLIB_PATH environmental variable to this directory.)
\$GLAS_HOME/cc_util	Contains makefile utilities.
\$GLAS_HOME/idl	Contains IDL code (if delivered)
\$GLAS_HOME/data	Contains sample control and ancillary files.
\$GLAS_HOME/src	Contains source code

### 4.2 Compilation

Change to the production directory and perform the following makes:

```
>cd $GLAS_HOME  
>make clean  
>make  
>make install
```

After this process is complete, all object files, libraries, and executables will have been re-created and installed in the appropriate directories. This is the recommended mode of compilation. It is the method used to create the distribution binaries and for final integration and acceptance testing.

#### **4.2.1 Debugging**

If debug output is desired, the software must be recompiled with the “debug” flag. Be aware this option generates a great deal of stdout output:

```
>cd $GLAS_HOME  
>make clean  
>make debug  
>make install
```

After this process is complete, all object files, libraries, and executables will have been re-created with the debug option and installed in the appropriate directories.

#### **4.2.2 Optimization**

If optimization is desired, the software must be recompiled with the “fast” flag.

```
>cd $GLAS_HOME  
>make clean  
>make fast  
>make install
```

After this process is complete, all object files, libraries, and executables will have been re-created, optimized with the “fast” option and installed in the appropriate directories.

**Section 5**  
**GLAS\_Exec**

GLAS\_Exec is responsible for controlling the data processing. It performs initializations, setting constants, reading ancillary data, data input and output, and global error handling. It also performs a synchronization function for processing the data granules.

GLAS\_Exec executes processes for the four major subsystems: L1A, Waveforms, Atmosphere, and Elevation. These subsystems contain implementations of the science algorithms used to process GLAS data

## **5.1 Startup and Termination**

This is an overview of the steps necessary to run GLAS\_Exec. Specific operational procedures for each of the supported V1 GLAS\_Exec scenarios will be provided in the ISIPS Operational Procedures Manual.

### **5.1.1 Setup a Runtime Directory**

The suggested method of running GLAS\_Exec is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a GLAS\_Exec run, one would perform the following steps (Designate TEMP\_DIR as the temporary directory and DATA\_DIR as the location in which input data for this job has been staged):

```
>mkdir $TEMP_DIR  
>cd $TEMP_DIR  
>ln -s $GLAS_HOME/bin/GLAS_Exec .  
>ln -s $GLAS_HOME/lib/* .  
>ln -s $GLAS_HOME/data/anc07*.dat .  
>ln -s $DATA_DIR/*.dat .
```

### **5.1.2 Create a Control File**

GSAS is designed to take the name of a control file as a command-line argument. All GSAS control files are in GSAS text-based “keyword=value” format. The GLAS\_Exec control file is fully documented in Appendix A. Sample control files are included as well.

The suggested method of creating a control file is by copying a template from the \$GLAS\_HOME/data directory and modifying it with desired input/output filenames and data processing options.

```
>cd $TEMP_DIR  
>cp $GLAS_HOME/data/control_template ./control_file_name  
>vi ./control_file_name
```

### 5.1.3 Run the Executable

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

```
>cd $TEMP_DIR  
>./GLAS_Exec control_file_name
```

### 5.1.4 Termination

The process will terminate automatically at the end of all input data. The log/meta-data file (ANC06\*) must be examined to determine runtime success. The ANC06 file is in GSAS text-based “keyword=value” format. The “value” field has two columns. The first is a timestamp, based on time of the data being processed. the second field is a message. ANC06 format and contents are documented in Appendix B. Sample ANC06 files are provided, as well.

The UNIX grep utility is useful for extracting such things as version numbers and error messages from the ANC06 file.

## 5.2 Functions and Their Operation

GLAS\_Exec primary function is to generate the GLAS Standard data products. GLAS\_Exec reads input GLA files record by record and generates the output data record by record. Ancillary data are read entirely by GLAS\_Exec during initialization when the ancillary data file is manageable enough to handle in core. Otherwise, the ancillary data are read as required by the subsystems. GLAS\_Exec requires a control file as an input argument and reports error and status messages via stdout and an ANC06 file.

GLAS\_Exec can perform different functions based on the contents of its control file. Processing options and required files for the supported scenarios are defined in the ISIPS Operational Procedures Manual. This subsection describes input and output files used by GLAS\_Exec and the four processing scenarios supported in V1.

### 5.2.1 Input/Output Files

The list of input and output file types is contained in Appendix A, as are recommended file naming conventions. See the GLAS Science Data Management Plan [Reference: Information Document 2.3d] for full details on the GLAS input and output files.

#### 5.2.1.1 Control Files

GLAS\_Exec receives control input via a control file. The control file may be generated by the SDMS or by hand. Each instance of a run will require a separate control file since version and file naming information is within the control file. The suggested method is to modify one of the supplied control file templates or “recipes”.

The control file name is passed as a command line argument to GLAS\_Exec. Appendix A details the format of a GLAS\_Exec control file and provides examples for each supported scenario.

### 5.2.1.2 GLAxx (Product) Files

The production of Product files is the primary task of GLAS\_Exec. Most Product files may be input or output, depending upon the scenario. Product files are integer-binary format and not human-readable. Utility software is provided to convert Product files into human-readable format. See the Product Readers/Writers section for details.

### 5.2.1.3 ANC06 Files

An ANC06 file will be generated for each execution of the GLAS\_Exec. The file will contain processing information and status, error messages, QA data, and data required to generate the GLAS product metadata. The ANC06 file is an ASCII file whose contents are in keyword = value format. The value is a time-stamped ASCII message. Users can use grep or another search mechanism to review the values of specific keywords in the ANC06 file

### 5.2.1.4 ANC07 Files

ANC07 files contain input error and status messages, as well as Science Team supplied constant values. The error and status messages are defined in the Error and Warning Message section. Contents of the constants ANC07 files are defined in Appendix B.

### 5.2.1.5 Other ANC Files

Specific processing scenarios may require other ancillary files. These files are fully documented in the GLAS Data Management Plan. Required files for specific scenarios are defined later in this section.

## 5.2.2 Supported Scenarios

There are several different processing scenarios defined for V1 GLAS\_Exec. This subsection will detail the required inputs and outputs for each scenario. Example control file templates and example control files are provided in Appendices C and D, respectively.

### 5.2.2.1 L1A Processing

For L1A processing, GLAS\_Exec reads GLA00 data, executes all of the L1A algorithms, and produces GLA01-03 (GLA04 will be delivered in V2).

#### Required Inputs:

- Control File
- GLA00 APIDs
- ANC31 Index File (in V2)
- ANC20 Predicted Orbit File
- ANC07 Errors
- ANC07 Global Constants
- ANC07 L1A Constants

#### Required Outputs:

GLA01 product files  
GLA02 product files  
GLA03 product files  
ANC06

**Required Processing Keywords:**

L1A\_PROCESS=ALL

**5.2.2.2 Waveform Processing**

For Waveform processing, GLAS\_Exec reads GLA01 data, executes all of the Waveform algorithms, and produces GLA05.

**Required Inputs:**

Control File  
GLA01  
ANC07 Errors  
ANC07 Global Constants  
ANC07 WF Constants

*and either*

ANC20 Predicted Orbit File

*or*

ANC08 Precision Orbit File  
ANC09 PAD File  
ANC24 ITRF matrix

**Required Outputs:**

GLA05 product files  
ANC06

**Required Processing Keywords:**

WF\_PROCESS=ALL

**5.2.2.3 Atmosphere Processing**

For Atmosphere processing, GLAS\_Exec reads GLA02 data, executes all of the Atmosphere algorithms, and produces GLA07-11.

**Required Inputs:**

Control File  
GLA02  
ANC01 Meteorological Data Files  
ANC12 DEM Files  
ANC07 Errors  
ANC07 Global Constants  
ANC07 Atmosphere Constants

ANC18 Standard Atmosphere File  
ANC30 Aerosol Categorization File  
ANC31 Troposphere Categorization File

*and either*

ANC20 Predicted Orbit File

*or*

ANC08 Precision Orbit File  
ANC09 PAD File  
ANC24 ITRF matrix

**Required Outputs:**

GLA07-11 product files  
ANC06

**Required Processing Keywords:**

ATM\_PROCESS=ALL

**5.2.2.4      Elevations Processing**

For Elevations processing, GLAS\_Exec reads GLA05 data, executes all of the Elevation algorithms, and produces GLA06 and GLA12-15.

**Required Inputs:**

Control File  
GLA05  
GLA07 (if available)  
ANC01 Meteorological Data File  
ANC20 Predicted Orbit File or ANC08 Precision Orbit File  
ANC09 PAD  
ANC12 Digital Elevation Model  
ANC13 Geoid File  
ANC24 ICRF to ITRF File  
ANC25 GPS to UTC File  
ANC16 Load Tide Model File  
ANC17 Ocean Tide Model File  
ANC19 Surface Type Class File  
ANC07 Errors  
ANC07 Global Constants  
ANC07 Elevation Constants

**Required Outputs:**

GLA06, 12-15 product files  
ANC06

**Required Processing Keywords:**

ELEV\_PROCESS=ALL  
SURF\_TYPE=ALL

**5.2.2.5 GLA05 to GLA06 Re-processing**

For L1B Elevation processing, GLAS\_Exec reads GLA05 data, executes the L1B Elevation algorithms, and produces GLA06.

**Required Inputs:**

Control File  
GLA05  
GLA07 (if available)  
ANC01 Meteorological Data File  
ANC12 Digital Elevation Model  
ANC13 Geoid File  
ANC25 GPS to UTC File  
ANC16 Load Tide Model File  
ANC17 Ocean Tide Model File  
ANC19 Surface Type Class File  
ANC07 Errors  
ANC07 Global Constants  
ANC07 Elevation Constants

*and either*

ANC20 Predicted Orbit File

*or*

ANC08 Precision Orbit File  
ANC09 PAD File  
ANC24 ITRF matrix

**Required Outputs:**

GLA06  
ANC06

**Required Processing Keywords:**

SURF\_TYPE=ALL  
ELEV\_PROCESS=ALL

**5.3 Error and Warning Messages**

See Appendix F.

## 5.4 Recovery Steps

If GLAS\_Exec terminates with an error:

- 1) Review error and status messages in the ANC06 file or on stdout to determine source or location of problem. Section 5.3 should aid in identifying and correcting specific problems.
- 2) Correct the problem
- 3) Remove previous output files
- 4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of GLAS\_Exec will be available for test use. This is, however, more of a developer procedure than a user procedure.



**Section 6**  
**Product Readers/Writers**

The GSAS product readers and writers are utility software delivered with GSAS. The team has made heavy use of these utilities during software testing. The primary function of these utilities is to read or write an integer-binary GLA file and print the contents of each record in human-readable format. Since these utilities are mostly used for debugging and not for standard GSAS production, they do not require a control file as input.

## **6.1 Startup and Termination**

This is an overview of the steps necessary to run the product readers. The writers use the same general steps. Differences are noted in the Functions and Operations section.

### **6.1.1 Setup a Runtime Directory**

The suggested method of running the product readers is to create a temporary directory and link all necessary files into it. For example, one would perform the following steps (Designate TEMP\_DIR as the temporary directory and DATA\_DIR as the location in which input data for this job has been staged):

```
>mkdir $TEMP_DIR  
>cd $TEMP_DIR  
>ln -s $GLAS_HOME/bin/glaxx_reader .  
>setenv SHLIB_PATH $GLAS_HOME/lib  
>chattr +s enable glaxx_reader  
>ln -s $DATA_DIR/*.dat .
```

### **6.1.2 Run the Executable**

The reader will interactively ask for an input filename and create output files based on the input name.

```
>cd $TEMP_DIR  
>./glaxx_reader
```

### **6.1.3 Termination**

The process will terminate automatically at the end of all input data. No ANC06 file is written. Any errors will appear on the screen (stdout).

## **6.2 Functions and Their Operation**

This section describes two different ways in which the readers/writers can be run and describes the input and output files.

## 6.2.1 Product Readers

There is a separate product reader executable for each GLA file type. In normal operation, the reader will read an integer-binary GLA file and print the contents of each record in human-readable format. A separate output file is created for product, algorithm, and scaling data.

Upon execution, the reader will interactively ask for an input filename. It will create output files based on the input file name as in Table 6-1:

**Table 6-1 Product Reader Output Files**

Filespec	Description
<i>inputname.read.alg</i>	Debug algorithm values.
<i>inputname.read.prod</i>	Debug hexadecimal product values
<i>inputname.read.scal</i>	Debug scaling values.

As a special function, each reader can be given a “-time” command line argument. In this case, the software requires an input file to be named glaxx\_timerun.out. The reader will then read the file without writing output in order to load-test the time it takes to read the input file.

### 6.2.1.1 Read a GLA Product File

The product readers can be used to read GLA product files and output their contents in human-readable format. The output files of the readers are described in the Users Guide section. There is a separate executable for each GLA product. We will refer to *xx* as the number of the specific product we wish to produce.

#### Inputs:

GLAxx product file.

#### Outputs:

Debug files (*name.read.prod*, *name.read.alg*, *name.read.scal*)

#### Steps:

Create and setup the runtime directory:

```
>mkdir $TEMP_DIR
>cd $TEMP_DIR
>ln -s $GLAS_HOME/bin/glaxx_reader .
>setenv SHLIB_PATH $GLAS_HOME/lib
>chatr +s enable glaxx_reader
```

Execute the binary and enter the name of the input file:

```
>./glaxx_reader
Enter name of input GLAxx file:
>myfile
Writing Scaling variables in myfile.read.scal
```

---

```
Writing Product variables in myfile.read.prod
Writing Algorithm variables in myfile.read.alg
```

Examine the debug output files (.scal, .prod, and .alg).

### 6.2.1.2 Example Load-test Product File Input

The product readers can be used to test how long it takes to read GLA product files. There are no intermediate output files in this scenario and the reader requires the input file be named glaxx\_timerun.out (This is normally the output of a product writer time test). There is a separate executable for each GLA product. We will refer to *xx* as the number of the specific product we wish to produce.

**Inputs:**

glaxx\_timerun.out

**Outputs:**

None.

**Steps:**

Create and setup the runtime directory:

```
>mkdir $TEMP_DIR
>cd $TEMP_DIR
>ln -s $GLAS_HOME/bin/glaxx_reader .
>setenv SHLIB_PATH $GLAS_HOME/lib
>chattr +s enable glaxx_reader
```

Use the unix time command to execute the binary. Use the -time command line argument:

```
>time ./glaxx_reader -time
```

The time command prints the time it took to read the (normally) 2883 records in the input file.

### 6.2.2 Product Writers

There is a separate product writer executable for each GLA file type. In normal operation, the writer will create an integer-binary GLA file and print the contents of each record in human-readable format. A separate output file is created for product, algorithm, and scaling data. Since these utilities are mostly used for debugging and not for standard GSAS production, they do not require a control file as input. Upon execution, the writer will interactively ask for an output filename. It will create output files based on the requested file name as shown in Table 6-2:

**Table 6-2 Product Writer Output Files**

Filespec	Description
<i>outputname</i>	Integer-binary GLA product file
<i>outputname.write.alg</i>	Debug algorithm values.

**Table 6-2 Product Writer Output Files**

<b>Filespec</b>	<b>Description</b>
<i>outputname.write.prod</i>	Debug hexadecimal product values
<i>outputname.write.scal</i>	Debug scaling values.

Each record of the integer-binary GLA product file is described in Table 6-3. The

**Table 6-3 Product Writer Output File Contents**

<b>Record Number</b>	<b>Description</b>
1	Initialization value.
2	Minimum value.
4- <i>n</i>	(Maximum-Minimum) / <i>n</i>
Last	Maximum value.

intention of the product writer is to create an output file with a linear range of values for each parameter from its minimum to maximum value, where *n* is the number of steps in the range. Normally, *n*=3. However, as a special function, each writer can be given a “-time” command line argument. In this case, the software will create an output file named glaxx\_timerun.out containing 2883 records of data. The reader will write the file without intermediate output in order to load-test the time it takes to write the output file. In this case, *n*=2880.

### 6.2.2.1 Example: Create a Sample GLA Product File

The product writers can be used to create sample GLA product files. The output files of the writers are described in the Users Guide section. There is a separate executable for each GLA product. We will refer to *xx* as the number of the specific product we wish to produce.

#### Inputs:

Name of output file.

#### Outputs:

GLAxx product file: *name.out*

Intermediate debug files (*name.write.prod*, *name.write.alg*, *name.write.scal*)

#### Steps:

Create and setup the runtime directory:

```
>mkdir $TEMP_DIR
>cd $TEMP_DIR
>ln -s $GLAS_HOME/bin/glaxx_writer .
```

```
>setenv SHLIB_PATH $GLAS_HOME/lib  
>chatr +s enable glaxx_writer
```

Execute the binary and enter the name of the output file:

```
>./glaxx_writer  
Enter name of output GLAxx file:  
>myfile  
Writing Scaling variables in myfile.write.scal  
Writing Product variables in myfile.write.prod  
Writing Algorithm variables in myfile.write.alg
```

Examine the debug output file (.scal, .prod, and .alg) to verify the product was written successfully.

#### 6.2.2.2 Example: Load-test Product File Output

The product writers can be used to test how long it takes to write GLA product files. There are no intermediate output files in this scenario. There is a separate executable for each GLA product. We will refer to *xx* as the number of the specific product we wish to produce.

**Inputs:**

None.

**Outputs:**

glaxx\_timerun.out

**Steps:**

Create and setup the runtime directory:

```
>mkdir TEMP_DIR  
>cd $TEMP_DIR  
>ln -s $GLAS_HOME/bin/glaxx_writer .  
>setenv SHLIB_PATH $GLAS_HOME/lib  
>chatr +s enable glaxx_writer
```

Use the unix time command to execute the binary. Use the -time command line argument:

```
>time ./glaxx_writer -time
```

The time command prints the time it took to write 2883 the records to the output file. The output file is named glaxx\_timerun.out.

### 6.3 Fault Recovery Procedures

- 1) Review error and status messages to determine source or location of problem.
- 2) Correct the problem.
- 3) Remove old output files.

- 4) Re-run the executable.

## **6.4 Emergency Procedures**

Not applicable in V1.

## **6.5 Diagnostic Procedures**

Use on-screen messages to determine execution problems. Reviewing the contents of any output files may be of help as well.

## Appendix A

# GLAS File Information

This Appendix lists the GLAS File types and provides information about the recommended production naming conventions.

### A.1 GLAS Production File Naming Convention

GSAS creates products and uses various ancillary files and other input products that are track/time dependent, or process-specific and created automatically. Some ancillary files are static and others are dynamic. The following scheme outlines the file naming conventions required for each file type.

- 1) Track dependent products processed at ISIPS (GLA01-3,05-15;BRW,QAP)

HHHxx\_mmm \_pr\_ccc\_tttt\_s \_nn\_ff.eee

- 2) Time dependent products: (GLA00 and GLA04, dynamic ANC,

HHHxx\_mmm yyyyymmdd\_hhmmss\_nn\_ff.eee

- 3) Process specific input or output automatically created and not covered under other categories (control files, log files, SCF qa)

HHHxx\_mmm \_yyyyymmdd\_i ii.eee  
)

- 4) Static Ancillary files required for processing (ANC)

HHHxx\_mmm\_nn\_ff.eee

Time for process-specific files is processing date. GLAS file naming keys are defined in Table 6-4.

**Table 6-4 GLAS File Naming Keys**

Key	Description
HHH	(Specific to 1,2,3 and 4) Type identification – GLA, ANC, SCF, BRW, QAP, ING, ...
HHH	(Specific to 3) Type of job - CF, ING, PRO, DIS, ...
xx	Type ID number (CCB assigned number within a specific HHH)
p	repeat ground track phase
r	reference orbit number
ccc	cycle of reference orbit for this phase
ttt	track within reference orbit

**Table 6-4 GLAS File Naming Keys (Continued)**

Key	Description
s	segment of orbit
yyyymmdd	starting date in year, month, and day of month
hhmmss	starting time hour, minute, second
mmm	release number for process that created the produce (CCB assigned-combination of software and data)
nn	granule version number (the number of times this granule is created for a specific release)
iii	counting sequence number (incremental sequence per day for each instance of a process specific HHHxx)
ff	file type (numerical, CCB assigned for multiple files as needed for data of same time period for a specific HHHxx, .i.e. multi-file granule)
eee	(specific to 1,2,4) file extension – dat, scf, hdf, eds, pds, met, ctl ...

## A.2 EDOS L0 File Naming Convention

EDOS-delivered files have a different naming convention than standard GLAS product files. The EDOS L0 files will be renamed as part of the ingest process. Incoming filenames are as follows:

```
foooooooottttthhhhhhdddddddiuu.eeee
```

Table 6-5 defines keys for the EDOS file naming convention.

**Table 6-5 EDOS L0 File Naming Keys**

Key	Description
f	E (for EDS) or P (for PDS) (1byte)
ooooooo	First APID in data set (SCID 3 bytes, APID 4 bytes)
tttttt	Second APID in data set (SCID 3 bytes, APID 4 bytes)
hhhhhhh	Third APID in data set (SCID 3 bytes, APID 4 bytes)
ddddddddd	creation GMT/ZULU time (11 bytes)
I	numeric identification (0-9) (1 byte)
uu	Unique file number (00-99) (2 bytes)
eeee	File naming extension PDS or EDS (4 bytes)

### A.3 GLAS File Type List

**Table 6-6 GLAS File List**

File ID	File Name	File Disposition
GLA00	GLAS Instrument Packet File	input
GLA01	Altimetry Data File	input/output
GLA02	Atmosphere Data File	input/output
GLA03	Engineering Data File	output
GLA04	SRS and GPS Data File	input/output
GLA05	Waveform-based Range Corrections File	input/output
GLA06	Elevation File	input/output
GLA07	Backscatter File	input/output
GLA08	Boundary Layer and Elevated Aerosol Layer Heights File	input/output
GLA09	Cloud Height for Multiple Layers File	input/output
GLA10	Aerosol Vertical Structure File	output
GLA11	Thin Cloud/Aerosol Optical Depth File	output
GLA12	Ice Sheet Products File	output
GLA13	Sea Ice Products File	output
GLA14	Land Products File	output
GLA15	Ocean Products File	output
ANC01	Meteorological Data File	input
ANC06	GLAS Metadata and Data Product Quality Data File	output
ANC07	GLAS Coefficients and Constants File	input
ANC08	Precision Orbit Data File	input
ANC09	Precision Attitude Data File	input
ANC12	Digital Elevation Model	input
ANC13	Geoid File	input
ANC16	Load Tide Model File	input
ANC17	Ocean Tide Model File	input
ANC18	Standard Atmosphere	input
ANC19	Surface Type Class File	input
ANC20	Predicted Orbit	input

**Table 6-6 GLAS File List**

ANC21	Science Trend Data	input
AN 22	Reference orbit file	input
ANC23	Reference ascending node file	input
ANC24	ICRF	input
ANC25	GPS-UTC time	input
ANC26	Ref orbit Ephemeris	input
ANC27	Regional mask	input
ANC28	Rev node table	input
ANC29	Index files	input/output
ANC30	Global aerosol categorization map (1 deg. By 1deg.)	input
ANC31	Aerosol tropospheric classification map (1 deg. By1 Deg.)	input
ANC32	Counter to UTC conversion file	input
ANC33	GPS time correction file	input

## A.4 Product Granule Sizes

GSAS software is not coded to work with specific granule sizes. However, in production, granules are defined to have fixed sizes, show in Table 6-7.

**Table 6-7 Product Granule Sizes**

Product	Level	Freq	Num Files	Rec Size	Granule Size	Granules/ Day	Revs/ Granule
GLA00	0	4/day	26	varies			
GLA01	1A	4/day	1	8040 b	30 MB	56	0.25
GLA02	1A	4/day	1	28652 b	354 MB	7	2
GLA03	1A	4/day	1	1000 b	86 MB	7	2
GLA04	1A	4/day	6	28000 b	346 MB	4	N/A
GLA05	1B	1/day	1	9636 b	15 MB	56	0.25
GLA06	1B	1/day	1	5236 b	8 MB	56	0.25
GLA07	1B	1/day	1	63836 b	788 MB	7	2
GLA08	2A	1/day	1	780 b	3 MB	1	14
GLA09	2A	1/day	1	2948 b	64 MB	1	14
GLA10	2A	1/day	1	13812 b	298 MB	1	14
GLA11	2B	1/day	1	568 b	12 MB	1	14

**Table 6-7 Product Granule Sizes (Continued)**

Product	Level	Freq	Num Files	Rec Size	Granule Size	Granules/ Day	Revs/ Granule
GLA12	2B	1/day	1	3004 b	29 MB	1	14
GLA13	2B	1/day	1	4076 b	18 MB	1	14
GLA14	2B	1/day	1	5416 b	166 MB	1	14
GLA15	2B	1/day	1	2888b	133 MB	1	14



## Appendix B

# ANC07 Format, Files and Contents

### B.1 ANC07 Format

ANC07 files are in standard GSAS “keyword=value” format. This format is:

```
[keyword]=[value]
```

The keyword is not case sensitive. Spaces are allowed, but not required. Comment lines must be prepended by a “#” character. The keyword is limited to 25 characters; the value is limited to 229 characters.

ANC07 files contain sections for errors messages, status messages, global constants, and subsystem-specific constants. Sections may be combined into a single file or separated into multiple files. Not all section need be present. The sections are delimited as follows:

```
BEG_OF_STATUS=
...Status section contents...
END_OF_STATUS

BEG_OF_ERROR=
...Error section contents...
END_OF_ERROR

BEG_OF_GLOBALS=
...Global constants section contents...
END_OF_GLOBALS

BEG_OF_ATM=
...Atmosphere constants section contents...
END_OF_ATM

BEG_OF_ELEV=
...Elevation constants section contents...
END_OF_ELEV

BEG_OF_L1A=
...L1A constants section contents...
END_OF_L1A

BEG_OF_UTIL=
...Utility constants section contents...
END_OF_UTIL
```

## B.2 ANC07 Files

In production, ANC07 is defined as a multi-granule file with a single granule representing a different section (for the most part). Table B-1 lists the granule names and contents. See Appendix A for more information regarding file naming conventions.

**Table B-1 ANC07 Granule Names and Contents**

Naming Convention	Contents
anc07_mmm_nn_00.dat	Error and Status sections
anc07_mmm_nn_01.dat	Global constants
anc07_mmm_nn_02.dat	Atmosphere constants
anc07_mmm_nn_03.dat	Elevation constants
anc07_mmm_nn_04.dat	Waveform constants
anc07_mmm_nn_05.dat	L1A constants
anc07_mmm_nn_06.dat	Utility constants

## B.3 ANC07 Contents

Global constants and those used by the L1A, Atmosphere, Elevation, and Waveforms are listed in this section. Error and Status messages are listed in Appendix E.

### B.3.1 L1A Constants

**Table B-2 L1A Constants**

Constants	Definition
GI_GRND_THRESH	ground ret threshold
GI_CLD_THRESH	cloud ret threshold #
GI_LID_SATVAL	lidar bin sat val
GI_INT_QFVAL	quality flag(1hz) for integrated return
GI_NRGMAX	laser energy max value #
GI_G_USEDBG	background index used for 532/1064
GI_IR_USEDBG	background index used for 532/1064
D_DTC_TBL1	deadtime correction lookup table
D_DTC_TBL2	
D_DTC_TBL3	
D_DTC_TBL4	
D_DTC_TBL5	
D_DTC_TBL6	
D_DTC_TBL7	
D_DTC_TBL8	
D_DTC_TBL9	
D_DTC_TBL10	
D_DTC_TBL11	
D_DTC_TBL12	
D_DTC_TBL13	
D_DTC_TBL14	
GD_G_MAXLID	deadtime correction lookup table
D_D2A_TBL1	digital to analog convrsn lookup table
D_D2A_TBL2	
D_D2A_TBL3	
D_D2A_TBL4	
D_D2A_TBL5	
D_D2A_TBL6	
D_D2A_TBL7	

**Table B-2 L1A Constants (Continued)**

<b>Constants</b>	<b>Definition</b>
D_D2A_TBL8	
D_D2A_TBL9	
D_D2A_TBL10	
D_D2A_TBL11	
D_D2A_TBL12	
D_D2A_TBL13	
D_D2A_TBL14	
GD_IR_MAXLID	digital to analog convrsn lookup table
I_POLDEG21	Number of coeffs in polynomial calculations for Apid21 (starting index is 16)
I_POLDEG	Number of coefficients in polynomial calculations for each lookup columns
D_OFFSET	offsets in polynomial calculations for the lookup table #
D_XA	First degree coefficients in polynomial calculations for the lookup table
I_LSRROSCT_N	Laser Oscillator Temperature(Celsius) (use lookup table column 1)
I_LSRDBLRT_N	Laser Doubler Temperature(Celsius)
I_LSRPST_N	Laser Power Supply Temperature (Celsius)
I_LSRPAT_N	Laser Preamp Temperature(Celsius)
I_532NRG_N	532 Energy (Percent) (use lookup table column 2)
I_DPINA_N	Dual Pin -A (Percent)
I_DPINB_N	Dual Pin -B (Percent)
I_PAD550V_N	Primary Altimeter Detector 550v(Volts) (use lookup table column 3)
I_SAD550V_N	Secondary Altimeter Detector 550v (Volts)
I_SPCMD1_N	SPCM Detector #1 550V (Volts)
I_SPCMD2_N	SPCM Detector #2 550V (Volts) #
I_SPCMD3_N	SPCM Detector #3 550V (Volts)
I_SPCMD4_N	SPCM Detector #4 550V (Volts)
I_SPCMD5_N	SPCM Detector #5 550V (Volts)
I_SPCMD6_N	SPCM Detector #6 550V (Volts)
I_SPCMD7_N	SPCM Detector #7 550V (Volts) #
I_SPCMD8_N	SPCM Detector #8 550V (Volts)

**Table B-2 L1A Constants (Continued)**

<b>Constants</b>	<b>Definition</b>
D_BA28INST_SF	28V Bus A Instrument (Volts) #
D_HYBSUPP_SF	Hybrid Supplies (Amps) #
D_HVPSDET_SF	HVPS Detector Supplies(Amps) #
D_OPHTRS_SF	Operational Heaters (Amps)
D_MECHSYS_SF	Mechanical System (Amps)
D_BB28L1V_SF	28V Bus B Laser 1 (Volts) #
D_BB28L1A_SF	28V Bus B Laser 1 (Amps)
D_BC28L2V_SF	28V Bus C Laser 2 (Volts)
D_BC28L2A_SF	28V Bus C Laser 2 (Amps)
D_BD28L3V_SF	28V Bus D Laser 3 (Volts) #
D_BD28L3A_SF	28V Bus D Laser 3 (Amps) #
D_5HYB1V_SF	5V Hybrid #1 (Volts) #
D_5HYB1A_SF	5V Hybrid #1 (Amps)
D_12VHYB2V_SF	12V Hybrid #2 (Volts)
D_12VHYB2A_SF	12V Hybrid #2 (Amps)
D_M12VHYB3V_SF	m12V Hybrid #3 (Volts)
D_M12VHYB3A_SF	m12V Hybrid #3 (Amps)
D_5VHYB4V_SF	5V Hybrid #4 (Volts)
D_5VHYB4A_SF	5V Hybrid #4 (Amps)
D_M5VHYB5V_SF	m5V Hybrid #5 (Volts)
D_M5VHYB5A_SF	m5V Hybrid #5 (Amps)
D_M5VHYB6V_SF	m5V Hybrid #6 (Volts)
D_M5VHYB6A_SF	m5V Hybrid #6 (Amps)
D_15VBPR_SF	15V Boost Post Reg (Volts)
D_M15VBPR_SF	m15V Boost Post Reg (Volts)
D_12VPOTC_SF	12V Prim Osc Thermal Control (Amps)
D_12VSOTC_SF	12V Sec Osc Thermal Control (Amps)
D_M2VDV_SF	m2V Discrete Voltage (Volts)
D_HBHS_SF	Hybrid Heatsink (deg C)
D_FETSBHS_SF	FET Switch Bank Heatsink (deg C)

**Table B-2 L1A Constants (Continued)**

<b>Constants</b>	<b>Definition</b>
D_HVPS0VR_SF	HVPS 0V Reference (Volts)
D_HVPS5VR_SF	HVPS 5V Reference (Volts)
I_HKBDC0_N	"Housekeeping Board Temperature, Ch 0 (use lookup table column 4)"#
I_INSPSBDC1_N	"Instrument Processor Sys Bord Temp, Ch1"
I_PCBDC2_N	"Photon Counter Board Temp, Ch2"
I_CDFTBDC3_N	"Cloud Digitizer/Frequency & Time Board Temp, Ch3"
I_AD1BDC4_N	"Altimeter Digitizer 1 Board Temp 1, Ch4"
I_AD1BDC5_N	"Altimeter Digitizer 1 Board Temp 1, Ch5"
I_AD2BDC6_N	"Altimeter Digitizer 2 Board Temp 1, Ch6"#
I_AD2BDC7_N	"Altimeter Digitizer 2 Board Temp 2, Ch7"
I_DCHBDC8_N	"Data Collection & Handling Board Temp, Ch8"
I_LMBDC9_N	"Laser Monitor Board Temp, Ch9"
I_TCMBDC10_N	"Temperature Controller Monitor Board Temp, Ch10"
I_OXCO1BC11_N	"Oven-crystal-controlled Oscillator 1 Board Temp, Ch11"
I_OXCO2BC12_N	"Oven-crystal-controlled Oscillator 2 Board Temp, Ch12"
I_OSCBDC13_N	Oscillator Board Temp, Ch13
I_OTSBD14_N	Optical Test Source Board Temp, Ch14#
I_LPAC15_N	Laser Profile Array Temp 1, Ch15#
I_LPAC16_N	Laser Profile Array Temp 2, Ch16#
I_AD1BDC17_N	Altimeter Digitizer 1 Board Temp 3, Ch17
I_AD2BDC18_N	Altimeter Digitizer 2 Board Temp 3, Ch18#
I_AD1BDC19_N	Altimeter Digitizer 1 Board Temp 4, Ch19
I_AD2BDC20_N	Altimeter Digitizer 2 Board Temp 4, Ch20#
I_AD1BDC21_N	Altimeter Digitizer 1 Board Temp 5, Ch21
I_AD2BDC22_N	Altimeter Digitizer 2 Board Temp 5, Ch22
I_PRTGYRO_N	PRT Gyro Temperature, Ch24 (use lookup table column 5)"
I_PRTSTCM_N	PRT, Star Camera Temperature, Ch25
I_PRTSRSC26_N	PRT, Stellar Reference System Temperature, Ch26
I_PRTLIDT_N	PRT, Lidar Detector Pkg Temperature, Ch27
I_PRTAD1_N	PRT, Altimeter Detector 1 Temperature, Ch28

**Table B-2 L1A Constants (Continued)**

<b>Constants</b>	<b>Definition</b>
I_PRTAD2_N	PRT, Altimeter Detector 2 Temperature, Ch29
I_PRTSCINF_N	PRT, Spacecraft Interface Temperature, Ch30
I_PRTTSMT_N	PRT, Telescope Mount Temperature, Ch31
I_PRTTSBF_N	PRT, Telescope Baffle Temperature, Ch32
I_PRTSR10_N	PRT, Temperature Region 10, Ch33, Spare
I_PRTSR11_N	PRT, Temperature Region 11, Ch34, Spare #
I_PRTCALL_N	PRT, Cal Low, Ch35
I_PRTCALH_N	PRT, Cal High, Ch36
I_PDBIASV_N	Pin Diode Bias Voltage (use lookup table column 6)##
I_TSR0PMIR_N	Telescope Region 0 primary Mirror (Celsius)(use lookup table column 7)
I_TSR1SMIR_N	Telescope Region 1 Secondary Mirror (Celsius)##
I_TSR2TWR_N	Telescope Region 2 Tower (Celsius)
I_ETC37I_N	Etalon Temperature, Ch37d (Celsius) (use lookup table column 8)"##
I_LHP1C37E_N	Loop Heat Pipe 1 Temperature, Ch37e (Celsius) (use lookup table column 9)##
I_LHP2C37F_N	Loop Heat Pipe 2 Temperature, Ch37f (Celsius)##
I_TSPMIRHD_N	Telescope Primary Mirror Heater Drive (Amps) (use lookup table column 10)##
I_TSSMIRHD_N	Telescope Secondary Mirror Heater Drive (Amps)
I_TSTWRHD_N	Telescope Tower Heater drive current (Amps)##
I_ETHCC37J_N	Etalon Heater Current, Ch37j (Amps)
I_HOPVC1_N	HOP and VC Thermistors - 1 (Celsius) (use lookup table column 11)
I_HOPVC2_N	HOP and VC Thermistors - 2 (Celsius)
I_HOPVC3_N	HOP and VC Thermistors - 3 (Celsius)##
I_HOPVC7_N	HOP and VC Thermistors - 7 (Celsius)
I_HOPVC8_N	HOP and VC Thermistors - 8 (Celsius)##
I_HOP1ACH1_N	HOP 1 Actuator Current - Heater 1 (Amps) (use lookup table column 12)
I_HOP2ACH1_N	HOP 2 Actuator Current - Heater 1 (Amps)
I_HOP3ACH1_N	HOP 3 Actuator Current - Heater 1 (Amps)##
I_HOP4ACH2_N	HOP 4 Actuator Current - Heater 2 (Amps)

**Table B-2 L1A Constants (Continued)**

Constants	Definition
I_HOP5ACH2_N	HOP 5 Actuator Current - Heater 2 (Amps)
I_HOP6ACH2_N	HOP 6 Actuator Current - Heater 2 (Amps)

### B.3.2 Elevation Constants

**Table B-3 Elevation Constants**

Constants	Definition
gravEffTide	gravity effect in milligals
earthTdH2	Constants used in calc of earth tide
earthTdH3	Constants used in calc of earth tide
DEMdLat	delta lat for DEM grid
DEMdLon	delta lon for DEM grid
DEMLAT_BEG	Start latitude on the DEM grid
DEMLON_BEG	Start longitude on the DEM grid #
REC_IMPULSE	Receiver Impulse Width (in ns)
DIVANGLE	Beam Divergence Angle
MAXPADOFF	Max PAD offset allowed (max off-nadir pointing angle)

### B.3.3 Waveform Constants

**Table B-4 Waveform Constants**

Waveform constants	Definition
D_PSAT_STOP1	Don't process WF if saturation .ge. d_psat_stop1
D_PSAT_STOP2	Don't process WF if saturation .ge. d_psat_stop2
D_PSAT_SPEC1	If d_psat_spec1 .ie. WF sat < d_psat_stop1 then use special processing
D_PSAT_SPEC2	If d_psat_spec2 .ie. WF sat < d_psat_stop2 then use special processing #
I_OFFSETB1	If i_slctRegn1 is set, indexSignalBegin1 indexSignalBegin1 - i_offsetb1#
I_OFFSETB2	If i_slctRegn2 is set, indexSignalBegin2 indexSignalBegin2 - i_offsetb2#
I_OFFSETE1	If i_slctRegn1 is set, indexSignalEnd1 indexSignalEnd1 + i_offsete1
I_OFFSETE2	If i_slctRegn2 is set, indexSignalEnd2 indexSignalEnd2 + i_offsete2#
I_MAXITER1	Max number of iterations during functional fit #
I_MAXITER2	Max number of iterations during functional fit
I_MAXFIT1	Max number of peaks to fit - land parameters
I_MAXFIT2	Max number of peaks to fit - other than land parameters #
D_NPEAK_MIN1	Min peak amplitude -> d_Npeak_min1 * (noise + sDevNoise) #
D_NPEAK_MIN2	#

**Table B-4 Waveform Constants**

Waveform constants	Definition
D_MINAMPPCNT1	Min peak amplitude -> d_minAmpPcnt1 * MaxAmp
D_MINAMPPCNT2	Min peak amplitude -> d_minAmpPcnt2 * MaxAmp
D_NSIG1	Noise threshhold -> d_nsig1 * sDevNoise
D_NSIG2	Noise threshhold : d_nsig1 * sDevNoise
D_INTV_MIN1	Min interval between peaks
D_INTV_MIN2	Min interval between peaks
I_SLCTREGN1	0:use all gates, 1:use selected region
I_SLCTREGN2	0:use all gates, 1:use selected region
D_FLTRWDMIN	Min filter width used by W_Smooth1
D_FLTRWDMAX	Max filter width used by W_Smooth1
D_DTHIRES	Time between gates for highest resolution #
D_CMB1	"(0,1) -> weight peaks by area when combining
D_CMB2	"(1,0) -> use straight average when combining peaks
D_SIGMAMINIT	Min peak sigma #
D_DNOISE1	
D_DNOISE2	
D_DAMPCK1	
D_DAMPCK2	
D_DTMCK1	
D_DTMCK2	
D_DSGMCK1	
D_DSGMCK2	
I_MINITER	Min number of iterations during functional fit
INGATES2USE1	Number of gates to use to calculate noise
INGATES2USE2	
IMINNG2USE	Min number of gates to use to calculate noise
I_NSCAL1	0:use observed noise & sDevNoise, 1:calculate noise & sDevNoise
I_NSCAL2	
I_MIN4SAT	Min number of maxAmp in a row to count as saturated

**Table B-4 Waveform Constants**

<b>Waveform constants</b>	<b>Definition</b>
D_THRESH_LVL1	retracker threshold : d_thresh_lvl * maxSmoothAmp(WF)
D_THRESH_LVL2	
D_V0NS1	a priori fit-matrix-sigma for noise : 1/SQRT(d_V0ns)
D_V0NS2	
D_V0AMP1	a priori fit-matrix-sigma for amplitude : 1/SQRT(d_V0amp)
D_V0AMP2	
D_V0LOC1	a priori fit-matrix-sigma for location : 1/SQRT(d_V0loc)
D_V0LOC2	5.0d0
D_V0SGM1	a priori fit-matrix-sigma for peak-sigma : 1/SQRT(d_V0sgm)
D_V0SGM2	
D_MINNOISE	Minimum noise

### B.3.4 Atmosphere Constants

**Table B-5 Atmosphere Constants**

Constants	Definition
GD_AER_20S_FTR	Aerosol scaling factor for 20 sec search #
GD_AER_4S_FTR	Aerosol scaling factor for 4 sec search
GD_BAD_PCT1	Percentage of bad records (in decimal form) for calib flag to be 1
GD_BAD_PCT2	Percentage of bad records (in decimal form) for calib flag to be 2#
GD_BMDIFF_FTR	Factor to multiply molecular difference for bot threshold for 20 s algm
GD_BTMR_FTR	Factor to multiply top threshold for bot threshold for 20 s algm
GD_CLD_PCT	Percentage of cloudy profiles (in decimal form) for calib flag to be 2
GD_CLD_THICK	Cloud thickness threshold #
GD_CLD THR	Threshold for cloudiness in 532 profile #
GD_CLD THR_FTR	Cloud threshold factor
GD_CRAT_THR	Critical ratio threshold for 4 s algm #
GD_DEM_HI	Upper limit for valid DEM (meters)
GD_DEM_LO	Lower limit for valid DEM (meters)
GD_DIURNAL_FTR	Diurnal threshold factor
GD_DNF THR1	Lower threshold for 532 day/night flag #
GD_DNF THR2	Upper threshold for 532 day/night flag
GD_G_LMCALCOF	Lab measured 532 calibration coefficient
GD_GDET_FTR	Ground detection factor #
GD_IR_LMCALCOF	Lab measured 1064 calibration coefficient #
GD_MAVG_FTR	Factor to multiply average molecular bscs for top threshold for 20 s algm
GD_NLAT	Northern latitude for calibration band (degrees North)
GD_PBL THR_FTR	PBL threshold factor #
GD_RAT THR1	Lower threshold for ratio of GLA07 integrated returns
GD_RAT THR2	Upper threshold for ratio of GLA07 integrated returns
GD_SEG_HTS	Segment heights for cloud detection (tops of bins)
GD_SLAT	Southern latitude for calibration band (degrees North)
GD_TIME_GAP	Number of seconds for a time gap (sec)
GD_TMDIFF_FTR	Factor to multiply molecular difference for top threshold for 20 s algm
GI_1SEC_MIN	Minimum number of valid 5 Hz profiles in 1 sec buffer

**Table B-5 Atmosphere Constants**

<b>Constants</b>	<b>Definition</b>
GI_20SEC_MIN	Min number of valid 5 Hz profiles in 20 sec buffer
GI_20SEC_MIN1	Minimum number of valid 1 sec profiles in 20 sec buffer
GI_4SEC_MIN	Min number of valid 5 Hz profiles in 4 sec buffer
GI_4SEC_MIN1	Minimum number of valid 1 sec profiles in 4 sec buffer
GI_4S_LIM_BIN	Number of bins to constrain the search for the PBL layer at 4 sec
GI_5HZ_LIM_BIN	Number of bins to constrain the search for the PBL layer at 5 Hz
GI_AER_BOT_HT	Height (m) above PBL or surface for aerosol search limit #
GI_AER_CLD_OFF	Number of bins above optically thick cloud top to end aerosol search at 4 sec #
GI_AER_LIM_BIN	Number of bins to constrain the search for aerosol top at 4 sec
GI_AER_NUM_BIN	Minimum number of bins in an aerosol layer in 20 sec algm
GI_AER_TOP_HT	Start Height (m) for 20 sec aerosol search limit
GI_BAD_LOC_LIMIT	Limit for number of bad locations allowed #
GI_BSCS_QF_CHK	Flag to indicate if backscatter quality flags are used to eliminate data. Not used : 0, used : 1
GI_CLD_BIN	Number of bins cld tops/bots must be within previous resolution's tops/bots
GI_CONS_AER20	Number of consecutive bins for aerosol layer at 20 sec
GI_CONS_AER4	Number of consecutive bins for aerosol layer at 4 sec
GI_CONS_GRD	Number of consecutive bins for ground detection #
GI_CONS_PBL	Number of consecutive bins for PBL
GI_DET_TOP_BIN	Number of bins above surface to start search for ground detection
GI_GDET_BIN	If ground detection is less than DEM by this number of bins, use DEM for lower limit of profile search
GI_NUM_CLD_HI	Number of consecutive bins for clouds from 10 to 20 km
GI_NUM_CLD_LO	Number of consecutive bins for clouds from 0 to 10 km
GI_NUM_CLR_HI	Number of consecutive bins for clear sky for 10 to 20 km
GI_NUM_CLR_LO	Number of consecutive bins for clear sky for 0 to 10 km
GI_PBL_CLD_HT	Backscatter profile is excluded from PBL search if a cloud top is detected above this height (m) + 20 sec average PBL height and no cloud bottom or ground is detected
GI_PBL_LIM1	Lower limit for input PBL height
GI_PBL_LIM2	Upper limit for input PBL height

**Table B-5 Atmosphere Constants**

<b>Constants</b>	<b>Definition</b>
GI_PBL_TOPHT	Height (m) above ground detection for PBL search
GI_PBL_TROPHT	Troposphere height (m) above average PBL height
GI_PSCHT	Height (m) above which polar stratospheric clouds are detected
GI_RN_BOT	Lower height (m) for random noise calculation
GI_RN_TOP	Upper height (m) for random noise calculation
GI_SEASON_DEMAR C	Day number of the year for start of Summer
GI_USE_GCC_F	Flag to use calculated or lab-measured constant 532 calib coeff; use constant value : 0; use calc high value : 1; calc low value : 2
GI_USE_IRCC_F	Flag to use calculated or lab-measured constant 1064 calib coeff; use constant value : 0; use calc low value : 1

**B.3.5 Global Constants****Table B-6 Global Constants**

<b>Constants</b>	<b>Definition</b>
ELLIPAE	Mean Ellipsoid Radius
ELLIPF	Ellipsoid Flattening

# Appendix C

## **GLAS\_Exec Control File Format and Elements**

### **C.1 Control File Format**

The GLAS\_Exec Control File is designed to be part of a larger control file used by one or more programs or utilities. As such, it specifies sections that identify the executable or functionality that will perform the task requiring the inputs contained in the section. Each section is bounded by an "=" sign in column 1, followed by the executable or task name that requires the control inputs.

Each section of the control file will contain lines will specify an input based on the 'KEYWORD=VALUE' construct. The construct consists of a line containing a keyword/value pair delimited by an equal sign (=). The ordering of the keywords is not relevant but should follow a convention for consistency. Multiple instances of certain keywords are allowed. The keyword is not case sensitive. Spaces are allowed, but not required. Comment lines must be prepended by a "#" character. The keyword is limited to 25 characters; the value is limited to 229 characters.

Required single-instance keywords include:

**Table C-1 Required Single-Instance Keywords**

<b>Keyword</b>	<b>Value</b>
TEMPLATE_NAME=	Name of the control file template.
EXEC_KEY=	Unique (per day) execution key
DATE_GENERATED=	Date the control file was generated.
OPERATOR=	Operator who generated the control file.

Optional multiple-instance keywords include:

**Table C-2 Optional Multiple-Instance Keywords**

<b>Keyword</b>	<b>Value</b>
CYCLE=	Cycle of data [number start_time stop_time]
REV=	Revolution of data [number start_time stop_time]
TRACK=	Track [number start_time stop_time]
INPUT_FILE=	Input file [filename start_time stop_time]
OUTPUT_FILE=	Output file [filename start_time stop_time]
SURFACE_TYPE=	Surface Type to Process (fixed values, see Table C-3)
L1A_PROCESS=	L1A Process to Execute (fixed values, see Table C-3)

**Table C-2 Optional Multiple-Instance Keywords (Continued)**

Keyword	Value
WAVEFORM_PROCESS=	Waveform Process to Execute (fixed values, see Table C-3)
ATMOSPHERE_PROCESS=	Atmosphere Process to Execute (fixed values, see Table C-3)
ELEVATION_PROCESS=	Elevation Process to Execute (fixed values, see Table C-3)

Additionally, pre-defined, subsystem-specific identifiers may specify which processes are executed, rather than a verbose list of processes. The SURFACE\_TYPE keyword specifies over what type of surface processing should occur. The default is all surfaces. Keywords and values are not case-specific (they will be converted to all lower case during parsing).

Keyword	Fixed Values
SURFACE_TYPE	ALL (default)
	LAND
	OCEAN
	SEAICE
	ICESHEET
L1A_PROCESS	ALL
	NONE (default)
	L_ALT
	L_ATM
	L_ENG

<b>Keyword</b>	<b>Fixed Values</b>
WAVEFORM_PROCESS	ALL
	NONE (default)
	W_Assess
	W_FunctionalFt
	W_doAllRgns
	W_processLand
	W_processOcean
	W_processIceS
	W_processSeal
	W_noGeo
WAVEFORM_OPTION	W_noLandCalc
	W_noOtherCalc
	w_doAllRgns
	w_processLand
	w_processOcean
	w_processIceS
	w_processSeal
ATMOSPHERE_PROCESS	w_noLandCalc
	w_noOtherCalc
	w_noGeo
	ALL
	NONE (default)
	A_no_pod
A_bs_to_end	A_bs_only
	A_cld_to_end
	A_bs_to_end

Keyword	Fixed Values
ELEVATION_PROCESS	ALL
	NONE (default)
	E_CalcLoadTD
	E_CalcOceanTD
	E_CalcEarthTD
	E_GetGeoid
	E_CalcTrop
	E_IntrpPOD
	E_CalcStdIR
	E_CalcLdIR
	E_CalcOclRc
	E_CalcSilR
	E_CalcIlsIR
	E_CalcSpLoc
	E_AtmQF
	E_CalcSlope
	E_CalcRefl
	E_ChckReg
	E_CalcRegRng
	E_CalcRegParm
	E_CalcDEM

Waveforms have additional flags beside the execution flags. These are options which control certain parameters within the waveform processing. Options in the Waveform subsystem processing are determined by the settings of the following flags:

**Table C-3 Waveform Control Flags**

Control	Result when Set=TRUE
I_doAllRgns	process frame data without checking region types mask
I_processLand	process frame data if any spot in the frame is within the land mask
I_processOcean	process frame data if any spot in the frame is within the ocean mask
I_processIceS	process frame data if any spot in the frame is within ice sheet mask
I_processSeal	process frame data if any spot in the frame is within sea ice mask

**Table C-3 Waveform Control Flags**

<b>Control</b>	<b>Result when Set=TRUE</b>
I_noLandCalc	do not process data using land parameters
I_noOtherCalc	do not process data using other-than-land parameters
I_noGeo	do not compute latitude, longitude, or elevation

## C.2 Start and Stop Time Designation

All files are required to be delimited by start and stop times. These times are currently floating point numbers. On both input and output, records are skipped until the time in the current record is greater than or equal to the specified start-time and less than or equal to the specified stop-time. Static ancillary files are required to have start-times and stop-times present, but these are currently ignored.

Files with INPUT\_FILE and OUTPUT\_FILE keywords must be listed in chronological order with respect to each keyword. This applies to CYCLE, TRACK, and REV keywords, as well.

## C.3 Required Control File Elements

As discussed previously, certain elements are required for all control files. Other elements are required for specific processing scenarios. This section will list templates with the requisite common elements as well as those elements specific to each of the four major processing scenarios. For other processing scenarios, subsets or combinations of the processing scenario-specific elements may be required.

### C.3.1 Common Control File Elements

```

#
#-----Start of Control File
#
# <description of control file>
#
#-----Execution Information
#
=GLAS_Exec
TEMPLATE=<template name>
EXEC_KEY=<unique execution key>
DATE_GENERATED=<date/time control file was generated>
OPERATOR=<operator name/uid>
#
#-----Cycle Track Boundaries
#
# Cycle Boundaries
CYCLE=<cycle_num starttime stoptime>
#
# Track Boundaries
TRACK=<track_num starttime stoptime>
#
# Rev Boundaries
REV=<rev_num starttime stoptime>
#
#-----Static ANC Files
#
# Common Input ANC07 Files: 00=error, 01=global
#
INPUT_FILE=<anc0700_error_filename starttime stoptime>
INPUT_FILE=<anc0701_global_filename starttime stoptime>
#
#-----Output ANC06 File
#
# Output ANC06
#
OUTPUT_FILE=<anc06_filename starttime stoptime>
```

#### *{processing scenario-specific elements}*

```

#
#-----End of Control File
#
=End of Control File
```

### C.3.2 L1A Processing-Specific Elements

```
#  
#-----Static ANC Files  
#  
# L1A Input ANC07 Files: 05=L1A  
#  
# INPUT_FILE=<anc0705_ll1a_filename starttime stoptime>  
#  
# Input DEM Files : 00 = DEM, 01=Mask  
#  
# INPUT_FILE=<anc1200_dem_filename starttime stoptime>  
INPUT_FILE=<anc1201_dem_filename starttime stoptime>  
#  
#-----Dynamic ANC Files  
#  
# Input Orbit  
#  
# INPUT_FILE=<orbit_filename starttime stoptime>  
#  
#-----Input gla Files  
#  
# Input gla00 Files  
#  
# INPUT_FILE=<gla00_filename starttime stoptime>  
#  
#-----Output GLA Files  
#  
# Output gla01-03 Files  
#  
# OUTPUT_FILE=<gla01_filename starttime stoptime>  
OUTPUT_FILE=<gla02_filename starttime stoptime>  
OUTPUT_FILE=<gla03_filename starttime stoptime>  
#  
#-----Execution Control  
#  
L1A_PROCESS=ALL
```

### C.3.3 Waveform Processing-Specific Elements

```
#  
-----Static ANC Files  
#  
# Input ANC07 Files : 04=waveform  
#  
INPUT_FILE=<anc0704_wf_filename starttime stoptime>  
#  
# Input DEM Files : 00 = DEM, 01=Mask  
#  
INPUT_FILE=<anc1200_dem_filename starttime stoptime>  
INPUT_FILE=<anc1201_dem_filename starttime stoptime>  
#  
# Input Geoid File  
#  
INPUT_FILE=<anc13_geoid_filename starttime stoptime>  
#  
-----Dynamic ANC Files  
#  
# Input POD - can be predicted (ANC20) or precision (ANC08) orbit file  
# If using precision (ANC08), the ANC24 is also required.  
#  
INPUT_FILE=<pod_filename starttime stoptime>  
#  
# Input PAD  
#  
INPUT_FILE=<pad_filename starttime stoptime>  
#  
# Input Rotation Matrix (optional, see Input POD note)  
#  
INPUT_FILE=<anc24_rot_filename starttime stoptime>  
#  
-----Input gla Files  
#  
# Input gla01 Files  
#  
INPUT_FILE=<gla01_filename starttime stoptime>  
#  
-----Output gla Files  
#  
# Output gla05 Files  
#  
OUTPUT_FILE=<gla05_filename starttime stoptime>  
#  
-----Output QAP Files  
#  
OUTPUT_FILE=<qap05_filename starttime stoptime>  
#  
-----Execution Control  
#  
WAVEFORM_PROCESS=ALL  
#
```

### C.3.4 Atmosphere Processing-Specific Elements

```
#  
#-----Static ANC Files  
#  
# Input ANC07 Files : 02=atmosphere  
#  
INPUT_FILE=<anc0702_atm_filename starttime stoptime>  
#  
# Input DEM Files : 00 = DEM, 01=Mask  
#  
INPUT_FILE=<anc1200_dem_filename starttime stoptime>  
INPUT_FILE=<anc1201_dem_filename starttime stoptime>  
#  
# Input Standard Atmosphere File  
#  
INPUT_FILE=<anc18_filename starttime stoptime>  
#  
# Input Global aerosol categorization map  
#  
INPUT_FILE=<anc30_filename starttime stoptime>  
#  
# INPUT Aerosol tropospheric classification map  
#  
INPUT_FILE=<anc31_filename starttime stoptime>  
#  
#-----Dynamic ANC Files  
#  
# Input POD - can be predicted (ANC20) or precision (ANC08) orbit file  
# If using precision (ANC08), the ANC24 is also required.  
#  
INPUT_FILE=<pod_filename starttime stoptime>  
#  
# Input PAD  
#  
INPUT_FILE=<pad_filename starttime stoptime>  
#  
# Input Rotation Matrix (optional, see Input POD note)  
#  
INPUT_FILE=<anc24_rot_filename starttime stoptime>  
#  
# Input MET Files : 00=hdr, 01=pwat, 02=hgt, 03=rh, 04=tmp  
#  
# 1st 6 hours  
INPUT_FILE=<anc0100_hdr_filename starttime stoptime>  
INPUT_FILE=<anc0101_pwat_filename starttime stoptime>  
INPUT_FILE=<anc0102_hgt_filename starttime stoptime>  
INPUT_FILE=<anc0103_rh_filename starttime stoptime>  
INPUT_FILE=<anc0104_tmp_filename starttime stoptime>  
#  
# 2nd 6 hours  
INPUT_FILE=<anc0100_hdr_filename starttime stoptime>  
INPUT_FILE=<anc0101_pwat_filename starttime stoptime>  
INPUT_FILE=<anc0102_hgt_filename starttime stoptime>
```

```
INPUT_FILE=<anc0103_rh_filename starttime stoptime>
INPUT_FILE=<anc0104_tmp_filename starttime stoptime>
#
#-----Input gla Files
#
# Input gla02 Files
#
INPUT_FILE=<gla02_filename starttime stoptime>
#
#-----Output gla Files
#
# Output gla07-11 Files
#
OUTPUT_FILE=<gla07_filename starttime stoptime>
OUTPUT_FILE=<gla08_filename starttime stoptime>
OUTPUT_FILE=<gla09_filename starttime stoptime>
OUTPUT_FILE=<gla10_filename starttime stoptime>
OUTPUT_FILE=<gla11_filename starttime stoptime>
#
#-----Output QAP Files
#
OUTPUT_FILE=<qap07_filename starttime stoptime>
#
#-----Execution Control
#
ATMOSPHERE_PROCESS=ALL
```

### C.3.5 Elevation Processing-Specific Elements

```
#  
#-----Static ANC Files  
#  
# Input ANC07 Files : 03=elevatoin  
#  
INPUT_FILE=<anc0703_elev_filename starttime stoptime>  
#  
# Input DEM Files : 00 = DEM, 01=Mask  
#  
INPUT_FILE=<anc1200_dem_filename starttime stoptime>  
INPUT_FILE=<anc1201_dem_filename starttime stoptime>  
#  
# Input Geoid File  
#  
INPUT_FILE=<anc13_filename starttime stoptime>  
#  
# Input Ocean and Load Tide Files  
#  
INPUT_FILE=<anc16_filename starttime stoptime>  
INPUT_FILE=<anc17_filename starttime stoptime>  
#  
#-----Dynamic ANC Files  
#  
# Input POD - can be predicted (ANC20) or precision (ANC08) orbit file  
# If using precision (ANC08), the ANC24 is also required.  
#  
INPUT_FILE=<pod_filename starttime stoptime>  
#  
# Input PAD  
#  
INPUT_FILE=<pad_filename starttime stoptime>  
#  
# Input Rotation Matrix (optional, see Input POD note)  
#  
INPUT_FILE=<anc24_rot_filename starttime stoptime>  
#  
#  
# Input MET Files : 00=hdr, 01=pwat, 02=hgt, 03=rh, 04=tmp  
#  
# 1st 6 hours  
INPUT_FILE=<anc0100_hdr_filename starttime stoptime>  
INPUT_FILE=<anc0101_pwat_filename starttime stoptime>  
INPUT_FILE=<anc0102_hgt_filename starttime stoptime>  
INPUT_FILE=<anc0103_rh_filename starttime stoptime>  
INPUT_FILE=<anc0104_tmp_filename starttime stoptime>  
#  
# 2nd 6 hours  
INPUT_FILE=<anc0100_hdr_filename starttime stoptime>  
INPUT_FILE=<anc0101_pwat_filename starttime stoptime>  
INPUT_FILE=<anc0102_hgt_filename starttime stoptime>  
INPUT_FILE=<anc0103_rh_filename starttime stoptime>  
INPUT_FILE=<anc0104_tmp_filename starttime stoptime>
```

```
#  
# Input ICRF to ITRF file  
#  
INPUT_FILE=<anc024_filename starttime stoptime>  
#  
# Input GPS to UTC file  
#  
INPUT_FILE=<anc025_filename starttime stoptime>  
  
-----Input gla Files  
#  
# Input gla05 Files  
#  
INPUT_FILE=<gla05_filename starttime stoptime>  
#  
-----Output gla Files  
#  
# Output gla06,12-15 Files  
#  
OUTPUT_FILE=<gla06_filename starttime stoptime>  
OUTPUT_FILE=<gla12_filename starttime stoptime>  
OUTPUT_FILE=<gla13_filename starttime stoptime>  
OUTPUT_FILE=<gla14_filename starttime stoptime>  
OUTPUT_FILE=<gla15_filename starttime stoptime>  
#  
-----Execution Control  
#  
ELEVATION_PROCESS=ALL
```

## Appendix D

# Sample Control Files

This appendix contains sample control files for the four major processing scenarios. These have been generated with appropriate granule sizes in mind.

### D.1 Sample L1A Control File

```
#  
#-----Start of Control File  
#  
# This is a sample L1A control file (1st of 4 segments/day)  
#  
#-----Execution Information  
#  
=GLAS_Exec  
#  
TEMPLATE=l1a  
EXEC_KEY=10023426  
DATE_GENERATED=19 April 2000  
OPERATOR=jlee  
#  
#-----Cycle Track Boundaries  
#  
# Cycle Boundaries  
#  
CYCLE=012 9201600 9225600  
#  
# Track Boundaries  
#  
TRACK= 1 9201600 9207600  
TRACK= 2 9207600 9213600  
TRACK= 3 9213600 9219600  
TRACK= 4 9219600 9225600  
#  
#-----Static ANC Files  
#  
# Input ANC07 Files : 00=error, 01=global, 05=L1A  
#  
INPUT_FILE=anc07_003_01_00.dat 9201600 9225600  
INPUT_FILE=anc07_003_01_01.dat 9201600 9225600  
INPUT_FILE=anc07_003_01_05.dat 9201600 9225600  
#  
# Predicted Orbit  
#  
INPUT_FILE=anc20_003_20000419_020725_00.dat 9201600 9219600  
#  
#-----Input GLA Files  
#  
# Input GLA00 Files
```

```

#
# ff: = corresponding APID
#
# 1st 6 hours
#
INPUT_FILE=GLA00_003_20000417000000_01_00.dat 9201600 9219600
INPUT_FILE=GLA00_003_20000417000000_01_13.dat 9201600 9219600
INPUT_FILE=GLA00_003_20000417000000_01_15.dat 9201600 9219600
INPUT_FILE=GLA00_003_20000417000000_01_17.dat 9201600 9219600
INPUT_FILE=GLA00_003_20000417000000_01_19.dat 9201600 9219600
INPUT_FILE=GLA00_003_20000417000000_01_20.dat 9201600 9219600
INPUT_FILE=GLA00_003_20000417000000_01_21.dat 9201600 9219600
INPUT_FILE=GLA00_003_20000417000000_01_22.dat 9201600 9219600
INPUT_FILE=GLA00_003_20000417000000_01_23.dat 9201600 9219600
INPUT_FILE=GLA00_003_20000417000000_01_24.dat 9201600 9219600
INPUT_FILE=GLA00_003_20000417000000_01_26.dat 9201600 9219600
#
# 2nd 6 hours
#
INPUT_FILE=GLA00_003_20000417060000_01_00.dat 9219600 9249600
INPUT_FILE=GLA00_003_20000417060000_01_13.dat 9219600 9249600
INPUT_FILE=GLA00_003_20000417060000_01_15.dat 9219600 9249600
INPUT_FILE=GLA00_003_20000417060000_01_17.dat 9219600 9249600
INPUT_FILE=GLA00_003_20000417060000_01_19.dat 9219600 9249600
INPUT_FILE=GLA00_003_20000417060000_01_20.dat 9219600 9249600
INPUT_FILE=GLA00_003_20000417060000_01_21.dat 9219600 9249600
INPUT_FILE=GLA00_003_20000417060000_01_22.dat 9219600 9249600
INPUT_FILE=GLA00_003_20000417060000_01_23.dat 9219600 9249600
INPUT_FILE=GLA00_003_20000417060000_01_24.dat 9219600 9249600
INPUT_FILE=GLA00_003_20000417060000_01_26.dat 9219600 9249600
#
#-----Output ANC Files
#
# Output ANC06
#
OUTPUT_FILE=anc06_003_20000419_012.dat 9201600 9225600
#
#-----Output GLA Files
#
# Output GLA01 Files
#
OUTPUT_FILE=GLA01_003_12_034_0001_1_00.dat 9201600 9203100
OUTPUT_FILE=GLA01_003_12_034_0001_2_00.dat 9203100 9204600
OUTPUT_FILE=GLA01_003_12_034_0001_3_00.dat 9204600 9206100
OUTPUT_FILE=GLA01_003_12_034_0001_4_00.dat 9206100 9207600
OUTPUT_FILE=GLA01_003_12_034_0002_1_00.dat 9207600 9209100
OUTPUT_FILE=GLA01_003_12_034_0002_2_00.dat 9209100 9210600
OUTPUT_FILE=GLA01_003_12_034_0002_3_00.dat 9210600 9212100
OUTPUT_FILE=GLA01_003_12_034_0002_4_00.dat 9212100 9213600
OUTPUT_FILE=GLA01_003_12_034_0003_1_00.dat 9213600 9215100
OUTPUT_FILE=GLA01_003_12_034_0003_2_00.dat 9215100 9216600
OUTPUT_FILE=GLA01_003_12_034_0003_3_00.dat 9216600 9218100
OUTPUT_FILE=GLA01_003_12_034_0003_4_00.dat 9218100 9219600
OUTPUT_FILE=GLA01_003_12_034_0004_1_00.dat 9219600 9221100

```

```
OUTPUT_FILE=GLA01_003_12_034_0004_2_00.dat 9221100 9222600
OUTPUT_FILE=GLA01_003_12_034_0004_3_00.dat 9222600 9224100
OUTPUT_FILE=GLA01_003_12_034_0004_4_00.dat 9224100 9225600
#
# Output GLA02 Files
#
OUTPUT_FILE=GLA02_003_12_034_0001_1_00.dat 9201600 9213600
OUTPUT_FILE=GLA02_003_12_034_0003_1_00.dat 9213600 9225600
#
# Output GLA03 Files
#
OUTPUT_FILE=GLA03_003_12_034_0001_1_00.dat 9201600 9225600
#
# Output GLA04 Files
# ff: 00=grs, 01=srs, 02=gyro, 03=TBD, 04=TDB
#
OUTPUT_FILE=GLA04_003_12_034_0003_1_00.dat 9201600 9225600
OUTPUT_FILE=GLA04_003_12_034_0003_1_01.dat 9201600 9225600
OUTPUT_FILE=GLA04_003_12_034_0003_1_02.dat 9201600 9225600
OUTPUT_FILE=GLA04_003_12_034_0003_1_03.dat 9201600 9225600
OUTPUT_FILE=GLA04_003_12_034_0003_1_04.dat 9201600 9225600
#
#-----Output QAP Files
#
# Output QAP01 Files
#
OUTPUT_FILE=QAP01_003_12_034_0001_1_00.dat 9201600 9203100
OUTPUT_FILE=QAP01_003_12_034_0001_2_00.dat 9203100 9204600
OUTPUT_FILE=QAP01_003_12_034_0001_3_00.dat 9204600 9206100
OUTPUT_FILE=QAP01_003_12_034_0001_4_00.dat 9206100 9207600
OUTPUT_FILE=QAP01_003_12_034_0002_1_00.dat 9207600 9209100
OUTPUT_FILE=QAP01_003_12_034_0002_2_00.dat 9209100 9210600
OUTPUT_FILE=QAP01_003_12_034_0002_3_00.dat 9210600 9212100
OUTPUT_FILE=QAP01_003_12_034_0002_4_00.dat 9212100 9213600
OUTPUT_FILE=QAP01_003_12_034_0003_1_00.dat 9213600 9215100
OUTPUT_FILE=QAP01_003_12_034_0003_2_00.dat 9215100 9216600
OUTPUT_FILE=QAP01_003_12_034_0003_3_00.dat 9216600 9218100
OUTPUT_FILE=QAP01_003_12_034_0003_4_00.dat 9218100 9219600
OUTPUT_FILE=QAP01_003_12_034_0004_1_00.dat 9219600 9221100
OUTPUT_FILE=QAP01_003_12_034_0004_2_00.dat 9221100 9222600
OUTPUT_FILE=QAP01_003_12_034_0004_3_00.dat 9222600 9224100
OUTPUT_FILE=QAP01_003_12_034_0004_4_00.dat 9224100 9225600
#
# Output QAP02 Files
#
OUTPUT_FILE=QAP02_003_12_034_0001_1_00.dat 9201600 9213600
OUTPUT_FILE=QAP02_003_12_034_0003_1_00.dat 9213600 9225600
#
# Output QAP03 Files
#
OUTPUT_FILE=QAP03_003_12_034_0001_1_00.dat 9201600 9225600
#
# Output QAP04 Files
# ff: 00=grs, 01=srs, 02=gyro, 03=TBD, 04=TDB
```

```
#  
OUTPUT_FILE=QAP04_003_12_034_0003_1_00.dat 9201600 9225600  
OUTPUT_FILE=QAP04_003_12_034_0003_1_01.dat 9201600 9225600  
OUTPUT_FILE=QAP04_003_12_034_0003_1_02.dat 9201600 9225600  
OUTPUT_FILE=QAP04_003_12_034_0003_1_03.dat 9201600 9225600  
OUTPUT_FILE=QAP04_003_12_034_0003_1_04.dat 9201600 9225600  
#  
#-----Execution Control  
#  
SURFACE_TYPE=ALL  
L1A_PROCESS=ALL  
#  
#-----End of Control File  
#  
=End of Control File
```

## D.2 Sample Waveform Control File

```
#  
-----Start of Control File  
#  
# This is a sample Waveform control file  
#  
-----Execution Information  
#  
=GLAS_Exec  
#  
TEMPLATE=waveform  
EXEC_KEY=10023424  
DATE_GENERATED=18 April 2000  
OPERATOR=jlee  
#  
-----Cycle Track Boundaries  
#  
# Cycle Boundaries  
#  
CYCLE=101 9201600 9285600  
#  
# Track Boundaries  
#  
TRACK= 1 9201600 9207600  
TRACK= 2 9207600 9213600  
TRACK= 3 9213600 9219600  
TRACK= 4 9219600 9225600  
TRACK= 5 9225600 9231600  
TRACK= 6 9231600 9237600  
TRACK= 7 9237600 9243600  
TRACK= 8 9243600 9249600  
TRACK= 9 9249600 9255600  
TRACK= 10 9255600 9261600  
TRACK= 11 9261600 9267600  
TRACK= 12 9267600 9273600  
TRACK= 13 9273600 9279600  
TRACK= 14 9279600 9285600  
#  
-----Static ANC Files  
#  
# Input ANC07 Files : 00=error, 01=global, 04=waveform  
#  
INPUT_FILE=anc07_003_01_00.dat 9201600 9285600  
INPUT_FILE=anc07_003_01_01.dat 9201600 9285600  
INPUT_FILE=anc07_003_01_04.dat 9201600 9285600  
#  
# Input DEM Files : 00 = DEM, 01=Mask  
#  
INPUT_FILE=anc12_003_01_00.dat 9201600 9285600  
INPUT_FILE=anc12_003_01_01.dat 9201600 9285600  
#  
# Input Geoid File  
#
```

```
INPUT_FILE=anc13_003_01_00.dat 9201600 9285600
#
# POD and PAD
#
INPUT_FILE=anc08_003_20000417_000000_00.dat 9201600 9285600
INPUT_FILE=anc09_003_20000417_000000_00.dat 9201600 9285600
INPUT_FILE=anc24_003_20000417_000000_00.dat 9201600 9285600
#
#-----Input GLA Files
#
# Input GLA01 Files
#
INPUT_FILE= GLA01_003_12_034_0001_1_00.dat 9201600 9203100
INPUT_FILE= GLA01_003_12_034_0001_2_00.dat 9203100 9204600
INPUT_FILE= GLA01_003_12_034_0001_3_00.dat 9204600 9206100
INPUT_FILE= GLA01_003_12_034_0001_4_00.dat 9206100 9207600
INPUT_FILE= GLA01_003_12_034_0002_1_00.dat 9207600 9209100
INPUT_FILE= GLA01_003_12_034_0002_2_00.dat 9209100 9210600
INPUT_FILE= GLA01_003_12_034_0002_3_00.dat 9210600 9212100
INPUT_FILE= GLA01_003_12_034_0002_4_00.dat 9212100 9213600
INPUT_FILE= GLA01_003_12_034_0003_1_00.dat 9213600 9215100
INPUT_FILE= GLA01_003_12_034_0003_2_00.dat 9215100 9216600
INPUT_FILE= GLA01_003_12_034_0003_3_00.dat 9216600 9218100
INPUT_FILE= GLA01_003_12_034_0003_4_00.dat 9218100 9219600
INPUT_FILE= GLA01_003_12_034_0004_1_00.dat 9219600 9221100
INPUT_FILE= GLA01_003_12_034_0004_2_00.dat 9221100 9222600
INPUT_FILE= GLA01_003_12_034_0004_3_00.dat 9222600 9224100
INPUT_FILE= GLA01_003_12_034_0004_4_00.dat 9224100 9225600
INPUT_FILE= GLA01_003_12_034_0005_1_00.dat 9225600 9227100
INPUT_FILE= GLA01_003_12_034_0005_2_00.dat 9227100 9228600
INPUT_FILE= GLA01_003_12_034_0005_3_00.dat 9228600 9230100
INPUT_FILE= GLA01_003_12_034_0005_4_00.dat 9230100 9231600
INPUT_FILE= GLA01_003_12_034_0006_1_00.dat 9231600 9233100
INPUT_FILE= GLA01_003_12_034_0006_2_00.dat 9233100 9234600
INPUT_FILE= GLA01_003_12_034_0006_3_00.dat 9234600 9236100
INPUT_FILE= GLA01_003_12_034_0006_4_00.dat 9236100 9237600
INPUT_FILE= GLA01_003_12_034_0007_1_00.dat 9237600 9239100
INPUT_FILE= GLA01_003_12_034_0007_2_00.dat 9239100 9240600
INPUT_FILE= GLA01_003_12_034_0007_3_00.dat 9240600 9242100
INPUT_FILE= GLA01_003_12_034_0007_4_00.dat 9242100 9243600
INPUT_FILE= GLA01_003_12_034_0008_1_00.dat 9243600 9245100
INPUT_FILE= GLA01_003_12_034_0008_2_00.dat 9245100 9246600
INPUT_FILE= GLA01_003_12_034_0008_3_00.dat 9246600 9248100
INPUT_FILE= GLA01_003_12_034_0008_4_00.dat 9248100 9249600
INPUT_FILE= GLA01_003_12_034_0009_1_00.dat 9249600 9251100
INPUT_FILE= GLA01_003_12_034_0009_2_00.dat 9251100 9252600
INPUT_FILE= GLA01_003_12_034_0009_3_00.dat 9252600 9254100
INPUT_FILE= GLA01_003_12_034_0009_4_00.dat 9254100 9255600
INPUT_FILE= GLA01_003_12_034_0010_1_00.dat 9255600 9257100
INPUT_FILE= GLA01_003_12_034_0010_2_00.dat 9257100 9258600
INPUT_FILE= GLA01_003_12_034_0010_3_00.dat 9258600 9260100
INPUT_FILE= GLA01_003_12_034_0010_4_00.dat 9260100 9261600
INPUT_FILE= GLA01_003_12_034_0011_1_00.dat 9261600 9263100
INPUT_FILE= GLA01_003_12_034_0011_2_00.dat 9263100 9264600
```

```
INPUT_FILE= GLA01_003_12_034_0011_3_00.dat 9264600 9266100
INPUT_FILE= GLA01_003_12_034_0011_4_00.dat 9266100 9267600
INPUT_FILE= GLA01_003_12_034_0012_1_00.dat 9267600 9269100
INPUT_FILE= GLA01_003_12_034_0012_2_00.dat 9269100 9270600
INPUT_FILE= GLA01_003_12_034_0012_3_00.dat 9270600 9272100
INPUT_FILE= GLA01_003_12_034_0012_4_00.dat 9272100 9273600
INPUT_FILE= GLA01_003_12_034_0013_1_00.dat 9273600 9275100
INPUT_FILE= GLA01_003_12_034_0013_2_00.dat 9275100 9276600
INPUT_FILE= GLA01_003_12_034_0013_3_00.dat 9276600 9278100
INPUT_FILE= GLA01_003_12_034_0013_4_00.dat 9278100 9279600
INPUT_FILE= GLA01_003_12_034_0014_1_00.dat 9279600 9281100
INPUT_FILE= GLA01_003_12_034_0014_2_00.dat 9281100 9282600
INPUT_FILE= GLA01_003_12_034_0014_3_00.dat 9282600 9284100
INPUT_FILE= GLA01_003_12_034_0014_4_00.dat 9284100 9285600
#
#-----Output ANC Files
#
# Output ANC06
#
OUTPUT_FILE=anc06_003_20000417_000.dat 9201600 9285600
#
#-----Output GLA Files
#
# Output GLA05 Files
#
OUTPUT_FILE= GLA05_003_12_034_0001_1_00.dat 9201600 9203100
OUTPUT_FILE= GLA05_003_12_034_0001_2_00.dat 9203100 9204600
OUTPUT_FILE= GLA05_003_12_034_0001_3_00.dat 9204600 9206100
OUTPUT_FILE= GLA05_003_12_034_0001_4_00.dat 9206100 9207600
OUTPUT_FILE= GLA05_003_12_034_0002_1_00.dat 9207600 9209100
OUTPUT_FILE= GLA05_003_12_034_0002_2_00.dat 9209100 9210600
OUTPUT_FILE= GLA05_003_12_034_0002_3_00.dat 9210600 9212100
OUTPUT_FILE= GLA05_003_12_034_0002_4_00.dat 9212100 9213600
OUTPUT_FILE= GLA05_003_12_034_0003_1_00.dat 9213600 9215100
OUTPUT_FILE= GLA05_003_12_034_0003_2_00.dat 9215100 9216600
OUTPUT_FILE= GLA05_003_12_034_0003_3_00.dat 9216600 9218100
OUTPUT_FILE= GLA05_003_12_034_0003_4_00.dat 9218100 9219600
OUTPUT_FILE= GLA05_003_12_034_0004_1_00.dat 9219600 9221100
OUTPUT_FILE= GLA05_003_12_034_0004_2_00.dat 9221100 9222600
OUTPUT_FILE= GLA05_003_12_034_0004_3_00.dat 9222600 9224100
OUTPUT_FILE= GLA05_003_12_034_0004_4_00.dat 9224100 9225600
OUTPUT_FILE= GLA05_003_12_034_0005_1_00.dat 9225600 9227100
OUTPUT_FILE= GLA05_003_12_034_0005_2_00.dat 9227100 9228600
OUTPUT_FILE= GLA05_003_12_034_0005_3_00.dat 9228600 9230100
OUTPUT_FILE= GLA05_003_12_034_0005_4_00.dat 9230100 9231600
OUTPUT_FILE= GLA05_003_12_034_0006_1_00.dat 9231600 9233100
OUTPUT_FILE= GLA05_003_12_034_0006_2_00.dat 9233100 9234600
OUTPUT_FILE= GLA05_003_12_034_0006_3_00.dat 9234600 9236100
OUTPUT_FILE= GLA05_003_12_034_0006_4_00.dat 9236100 9237600
OUTPUT_FILE= GLA05_003_12_034_0007_1_00.dat 9237600 9239100
OUTPUT_FILE= GLA05_003_12_034_0007_2_00.dat 9239100 9240600
OUTPUT_FILE= GLA05_003_12_034_0007_3_00.dat 9240600 9242100
OUTPUT_FILE= GLA05_003_12_034_0007_4_00.dat 9242100 9243600
OUTPUT_FILE= GLA05_003_12_034_0008_1_00.dat 9243600 9245100
```

```

OUTPUT_FILE= GLA05_003_12_034_0008_2_00.dat 9245100 9246600
OUTPUT_FILE= GLA05_003_12_034_0008_3_00.dat 9246600 9248100
OUTPUT_FILE= GLA05_003_12_034_0008_4_00.dat 9248100 9249600
OUTPUT_FILE= GLA05_003_12_034_0009_1_00.dat 9249600 9251100
OUTPUT_FILE= GLA05_003_12_034_0009_2_00.dat 9251100 9252600
OUTPUT_FILE= GLA05_003_12_034_0009_3_00.dat 9252600 9254100
OUTPUT_FILE= GLA05_003_12_034_0009_4_00.dat 9254100 9255600
OUTPUT_FILE= GLA05_003_12_034_0010_1_00.dat 9255600 9257100
OUTPUT_FILE= GLA05_003_12_034_0010_2_00.dat 9257100 9258600
OUTPUT_FILE= GLA05_003_12_034_0010_3_00.dat 9258600 9260100
OUTPUT_FILE= GLA05_003_12_034_0010_4_00.dat 9260100 9261600
OUTPUT_FILE= GLA05_003_12_034_0011_1_00.dat 9261600 9263100
OUTPUT_FILE= GLA05_003_12_034_0011_2_00.dat 9263100 9264600
OUTPUT_FILE= GLA05_003_12_034_0011_3_00.dat 9264600 9266100
OUTPUT_FILE= GLA05_003_12_034_0011_4_00.dat 9266100 9267600
OUTPUT_FILE= GLA05_003_12_034_0012_1_00.dat 9267600 9269100
OUTPUT_FILE= GLA05_003_12_034_0012_2_00.dat 9269100 9270600
OUTPUT_FILE= GLA05_003_12_034_0012_3_00.dat 9270600 9272100
OUTPUT_FILE= GLA05_003_12_034_0012_4_00.dat 9272100 9273600
OUTPUT_FILE= GLA05_003_12_034_0013_1_00.dat 9273600 9275100
OUTPUT_FILE= GLA05_003_12_034_0013_2_00.dat 9275100 9276600
OUTPUT_FILE= GLA05_003_12_034_0013_3_00.dat 9276600 9278100
OUTPUT_FILE= GLA05_003_12_034_0013_4_00.dat 9278100 9279600
OUTPUT_FILE= GLA05_003_12_034_0014_1_00.dat 9279600 9281100
OUTPUT_FILE= GLA05_003_12_034_0014_2_00.dat 9281100 9282600
OUTPUT_FILE= GLA05_003_12_034_0014_3_00.dat 9282600 9284100
OUTPUT_FILE= GLA05_003_12_034_0014_4_00.dat 9284100 9285600
#
#-----Output QAP Files
#
# Output QAP05 Files
#
OUTPUT_FILE= QAP05_003_12_034_0001_1_00.dat 9201600 9203100
OUTPUT_FILE= QAP05_003_12_034_0001_2_00.dat 9203100 9204600
OUTPUT_FILE= QAP05_003_12_034_0001_3_00.dat 9204600 9206100
OUTPUT_FILE= QAP05_003_12_034_0001_4_00.dat 9206100 9207600
OUTPUT_FILE= QAP05_003_12_034_0002_1_00.dat 9207600 9209100
OUTPUT_FILE= QAP05_003_12_034_0002_2_00.dat 9209100 9210600
OUTPUT_FILE= QAP05_003_12_034_0002_3_00.dat 9210600 9212100
OUTPUT_FILE= QAP05_003_12_034_0002_4_00.dat 9212100 9213600
OUTPUT_FILE= QAP05_003_12_034_0003_1_00.dat 9213600 9215100
OUTPUT_FILE= QAP05_003_12_034_0003_2_00.dat 9215100 9216600
OUTPUT_FILE= QAP05_003_12_034_0003_3_00.dat 9216600 9218100
OUTPUT_FILE= QAP05_003_12_034_0003_4_00.dat 9218100 9219600
OUTPUT_FILE= QAP05_003_12_034_0004_1_00.dat 9219600 9221100
OUTPUT_FILE= QAP05_003_12_034_0004_2_00.dat 9221100 9222600
OUTPUT_FILE= QAP05_003_12_034_0004_3_00.dat 9222600 9224100
OUTPUT_FILE= QAP05_003_12_034_0004_4_00.dat 9224100 9225600
OUTPUT_FILE= QAP05_003_12_034_0005_1_00.dat 9225600 9227100
OUTPUT_FILE= QAP05_003_12_034_0005_2_00.dat 9227100 9228600
OUTPUT_FILE= QAP05_003_12_034_0005_3_00.dat 9228600 9230100
OUTPUT_FILE= QAP05_003_12_034_0005_4_00.dat 9230100 9231600
OUTPUT_FILE= QAP05_003_12_034_0006_1_00.dat 9231600 9233100
OUTPUT_FILE= QAP05_003_12_034_0006_2_00.dat 9233100 9234600

```

```
OUTPUT_FILE= QAP05_003_12_034_0006_3_00.dat 9234600 9236100
OUTPUT_FILE= QAP05_003_12_034_0006_4_00.dat 9236100 9237600
OUTPUT_FILE= QAP05_003_12_034_0007_1_00.dat 9237600 9239100
OUTPUT_FILE= QAP05_003_12_034_0007_2_00.dat 9239100 9240600
OUTPUT_FILE= QAP05_003_12_034_0007_3_00.dat 9240600 9242100
OUTPUT_FILE= QAP05_003_12_034_0007_4_00.dat 9242100 9243600
OUTPUT_FILE= QAP05_003_12_034_0008_1_00.dat 9243600 9245100
OUTPUT_FILE= QAP05_003_12_034_0008_2_00.dat 9245100 9246600
OUTPUT_FILE= QAP05_003_12_034_0008_3_00.dat 9246600 9248100
OUTPUT_FILE= QAP05_003_12_034_0008_4_00.dat 9248100 9249600
OUTPUT_FILE= QAP05_003_12_034_0009_1_00.dat 9249600 9251100
OUTPUT_FILE= QAP05_003_12_034_0009_2_00.dat 9251100 9252600
OUTPUT_FILE= QAP05_003_12_034_0009_3_00.dat 9252600 9254100
OUTPUT_FILE= QAP05_003_12_034_0009_4_00.dat 9254100 9255600
OUTPUT_FILE= QAP05_003_12_034_0010_1_00.dat 9255600 9257100
OUTPUT_FILE= QAP05_003_12_034_0010_2_00.dat 9257100 9258600
OUTPUT_FILE= QAP05_003_12_034_0010_3_00.dat 9258600 9260100
OUTPUT_FILE= QAP05_003_12_034_0010_4_00.dat 9260100 9261600
OUTPUT_FILE= QAP05_003_12_034_0011_1_00.dat 9261600 9263100
OUTPUT_FILE= QAP05_003_12_034_0011_2_00.dat 9263100 9264600
OUTPUT_FILE= QAP05_003_12_034_0011_3_00.dat 9264600 9266100
OUTPUT_FILE= QAP05_003_12_034_0011_4_00.dat 9266100 9267600
OUTPUT_FILE= QAP05_003_12_034_0012_1_00.dat 9267600 9269100
OUTPUT_FILE= QAP05_003_12_034_0012_2_00.dat 9269100 9270600
OUTPUT_FILE= QAP05_003_12_034_0012_3_00.dat 9270600 9272100
OUTPUT_FILE= QAP05_003_12_034_0012_4_00.dat 9272100 9273600
OUTPUT_FILE= QAP05_003_12_034_0013_1_00.dat 9273600 9275100
OUTPUT_FILE= QAP05_003_12_034_0013_2_00.dat 9275100 9276600
OUTPUT_FILE= QAP05_003_12_034_0013_3_00.dat 9276600 9278100
OUTPUT_FILE= QAP05_003_12_034_0013_4_00.dat 9278100 9279600
OUTPUT_FILE= QAP05_003_12_034_0014_1_00.dat 9279600 9281100
OUTPUT_FILE= QAP05_003_12_034_0014_2_00.dat 9281100 9282600
OUTPUT_FILE= QAP05_003_12_034_0014_3_00.dat 9282600 9284100
OUTPUT_FILE= QAP05_003_12_034_0014_4_00.dat 9284100 9285600
#
#-----Execution Control
#
SURFACE_TYPE=ALL
WAVEFORM_PROCESS=ALL
#
=End of Control File
```

### D.3 Sample Atmosphere Control File

```

#
#-----Start of Control File
#
# This is a sample L1B and 2 Atmosphere control file
#
#-----Execution Information
#
=GLAS_Exec
#
TEMPLATE=l1a_and_2_atmosphere
EXEC_KEY=10023425
DATE_GENERATED=18 April 2000
OPERATOR=jlee
#
#-----Cycle Track Boundaries
#
# Cycle Boundaries
#
CYCLE=101 9201600 9285600
#
# Track Boundaries
#
TRACK= 1 9201600 9207600
TRACK= 2 9207600 9213600
TRACK= 3 9213600 9219600
TRACK= 4 9219600 9225600
TRACK= 5 9225600 9231600
TRACK= 6 9231600 9237600
TRACK= 7 9237600 9243600
TRACK= 8 9243600 9249600
TRACK= 9 9249600 9255600
TRACK= 10 9255600 9261600
TRACK= 11 9261600 9267600
TRACK= 12 9267600 9273600
TRACK= 13 9273600 9279600
TRACK= 14 9279600 9285600
#
#-----Static ANC Files
#
# Input ANC07 Files : 00=error, 01=global, 02=atmosphere
#
INPUT_FILE=anc07_003_01_00.dat 9201600 9285600
INPUT_FILE=anc07_003_01_01.dat 9201600 9285600
INPUT_FILE=anc07_003_01_02.dat 9201600 9285600
#
# Input DEM Files : 00 = DEM, 01=Mask
#
INPUT_FILE=anc12_003_01_00.dat 9201600 9285600
INPUT_FILE=anc12_003_01_01.dat 9201600 9285600
#
# Input Standard Atmosphere File
#

```

```
INPUT_FILE=anc18_003_01_00.dat 9201600 9285600
#
# Input Global aerosol categorization map
#
INPUT_FILE=anc30_003_01_00.dat 9201600 9285600
#
# INPUT Aerosol tropospheric classification map
#
INPUT_FILE=anc31_003_01_00.dat 9201600 9285600
#
#-----Dynamic ANC Files
#
# Input MET Files : 00 = hdr, 01=pwat, 02=hgt, 03=rh, 04=tmp
#
# 1st 6 hours
#
INPUT_FILE=anc01_003_20000417_000000_01_00.dat 9201600 9239100
INPUT_FILE=anc01_003_20000417_000000_01_01.dat 9201600 9239100
INPUT_FILE=anc01_003_20000417_000000_01_02.dat 9201600 9239100
INPUT_FILE=anc01_003_20000417_000000_01_03.dat 9201600 9239100
INPUT_FILE=anc01_003_20000417_000000_01_04.dat 9201600 9239100
#
# 2nd 6 hours
#
INPUT_FILE=anc01_003_20000417_120000_01_00.dat 9239100 9285600
INPUT_FILE=anc01_003_20000417_120000_01_01.dat 9239100 9285600
INPUT_FILE=anc01_003_20000417_120000_01_02.dat 9239100 9285600
INPUT_FILE=anc01_003_20000417_120000_01_03.dat 9239100 9285600
INPUT_FILE=anc01_003_20000417_120000_01_04.dat 9239100 9285600
#
# POD
#
INPUT_FILE=anc08_003_20000417_000000_00.dat 9201600 9285600
INPUT_FILE=anc09_003_20000417_000000_00.dat 9201600 9285600
INPUT_FILE=anc24_003_20000417_000000_00.dat 9201600 9285600
#
#-----Input GLA Files
#
# Input GLA02 Files
#
INPUT_FILE=GLA02_003_12_034_0001_1_00.dat 9201600 9213600
INPUT_FILE=GLA02_003_12_034_0003_1_00.dat 9213600 9225600
INPUT_FILE=GLA02_003_12_034_0005_1_00.dat 9225600 9237600
INPUT_FILE=GLA02_003_12_034_0007_1_00.dat 9237600 9249600
INPUT_FILE=GLA02_003_12_034_0009_1_00.dat 9249600 9261600
INPUT_FILE=GLA02_003_12_034_0011_1_00.dat 9261600 9273600
INPUT_FILE=GLA02_003_12_034_0013_1_00.dat 9273600 9285600
#
#-----Output ANC Files
#
# Output ANC06
#
OUTPUT_FILE=anc06_003_20000417_001.dat 0 9201600 9285600
#
```

```
#-----Output GLA Files
#
# Output GLA07 Files
#
OUTPUT_FILE=GLA07_003_12_034_0001_1_00.dat 9201600 9213600
OUTPUT_FILE=GLA07_003_12_034_0003_1_00.dat 9213600 9225600
OUTPUT_FILE=GLA07_003_12_034_0005_1_00.dat 9225600 9237600
OUTPUT_FILE=GLA07_003_12_034_0007_1_00.dat 9237600 9249600
OUTPUT_FILE=GLA07_003_12_034_0009_1_00.dat 9249600 9261600
OUTPUT_FILE=GLA07_003_12_034_0011_1_00.dat 9261600 9273600
OUTPUT_FILE=GLA07_003_12_034_0013_1_00.dat 9273600 9285600
#
# Output GLA08-11
#
OUTPUT_FILE= GLA08_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE= GLA09_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE= GLA10_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE= GLA11_003_12_034_0001_1_00.dat 9201600 9285600
#
#-----Output QAP Files
#
# Output QAP07 Files
#
OUTPUT_FILE=QAP07_003_12_034_0001_1_00.dat 9201600 9213600
OUTPUT_FILE=QAP07_003_12_034_0003_1_00.dat 9213600 9225600
OUTPUT_FILE=QAP07_003_12_034_0005_1_00.dat 9225600 9237600
OUTPUT_FILE=QAP07_003_12_034_0007_1_00.dat 9237600 9249600
OUTPUT_FILE=QAP07_003_12_034_0009_1_00.dat 9249600 9261600
OUTPUT_FILE=QAP07_003_12_034_0011_1_00.dat 9261600 9273600
OUTPUT_FILE=QAP07_003_12_034_0013_1_00.dat 9273600 9285600
#
# Output QAP08-11
#
OUTPUT_FILE=QAP08_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE=QAP09_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE=QAP10_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE=QAP11_003_12_034_0001_1_00.dat 9201600 9285600
#
#-----Execution Control
#
ATMOSPHERE_PROCESS=ALL
#
#-----End of Control File
#
=End of Control File
```

## D.4 Sample Elevation Control File

```
#  
-----Start of Control File  
#  
# This is a sample Elevation control file  
#  
-----Execution Information  
#  
=GLAS_Exec  
#  
TEMPLATE=Elevation  
EXEC_KEY=10023424  
DATE_GENERATED=18 April 2000  
OPERATOR=jlee  
#  
-----Cycle Track Boundaries  
#  
# Cycle Boundaries  
#  
CYCLE=101 9201600 9285600  
#  
# Track Boundaries  
#  
TRACK= 1 9201600 9207600  
TRACK= 2 9207600 9213600  
TRACK= 3 9213600 9219600  
TRACK= 4 9219600 9225600  
TRACK= 5 9225600 9231600  
TRACK= 6 9231600 9237600  
TRACK= 7 9237600 9243600  
TRACK= 8 9243600 9249600  
TRACK= 9 9249600 9255600  
TRACK= 10 9255600 9261600  
TRACK= 11 9261600 9267600  
TRACK= 12 9267600 9273600  
TRACK= 13 9273600 9279600  
TRACK= 14 9279600 9285600  
#  
-----Static ANC Files  
#  
# Input ANC07 Files : 00=error, 01=global, 03=elevation  
#  
INPUT_FILE=anc07_003_01_00.dat 9201600 9285600  
INPUT_FILE=anc07_003_01_01.dat 9201600 9285600  
INPUT_FILE=anc07_003_01_03.dat 9201600 9285600  
#  
# Input DEM Files : 00 = DEM, 01=Mask  
#  
INPUT_FILE=anc12_003_01_00.dat 9201600 9285600  
INPUT_FILE=anc12_003_01_01.dat 9201600 9285600  
#  
# Input Geoid File  
#
```

```
INPUT_FILE=anc13_003_01_00.dat 9201600 9285600
#
# Load and Ocean Tide Files: 16=load, 17=ocean
#
INPUT_FILE=anc16_003_01_00.dat 9201600 9285600
INPUT_FILE=anc17_003_01_00.dat 9201600 9285600
#
#-----Dynamic ANC Files
#
# Input MET Files : 00 = hdr, 01=pwat, 02=hgt, 03=rh, 04=tmp
#
# 1st 12 hours
#
INPUT_FILE=anc01_003_20000417_000000_01_00.dat 9201600 9239100
INPUT_FILE=anc01_003_20000417_000000_01_01.dat 9201600 9239100
INPUT_FILE=anc01_003_20000417_000000_01_02.dat 9201600 9239100
INPUT_FILE=anc01_003_20000417_000000_01_03.dat 9201600 9239100
INPUT_FILE=anc01_003_20000417_000000_01_04.dat 9201600 9239100
#
# 2nd 12 hours
#
INPUT_FILE=anc01_003_20000417_120000_01_00.dat 9239100 9285600
INPUT_FILE=anc01_003_20000417_120000_01_01.dat 9239100 9285600
INPUT_FILE=anc01_003_20000417_120000_01_02.dat 9239100 9285600
INPUT_FILE=anc01_003_20000417_120000_01_03.dat 9239100 9285600
INPUT_FILE=anc01_003_20000417_120000_01_04.dat 9239100 9285600
#
# POD and PAD
#
INPUT_FILE=anc08_003_20000417_000000_00.dat 9201600 9285600
INPUT_FILE=anc09_003_20000417_000000_00.dat 9201600 9285600
INPUT_FILE=anc24_003_20000417_000000_00.dat 9201600 9285600
#
# Input GPS to UTC Time file
#
INPUT_FILE=anc25_003_20000417_000000_00.dat 9201600 9285600
#
#
#-----Input GLA Files
#
# Input GLA05 Files
#
INPUT_FILE= GLA05_003_12_034_0001_1_00.dat 9201600 9203100
INPUT_FILE= GLA05_003_12_034_0001_2_00.dat 9203100 9204600
INPUT_FILE= GLA05_003_12_034_0001_3_00.dat 9204600 9206100
INPUT_FILE= GLA05_003_12_034_0001_4_00.dat 9206100 9207600
INPUT_FILE= GLA05_003_12_034_0002_1_00.dat 9207600 9209100
INPUT_FILE= GLA05_003_12_034_0002_2_00.dat 9209100 9210600
INPUT_FILE= GLA05_003_12_034_0002_3_00.dat 9210600 9212100
INPUT_FILE= GLA05_003_12_034_0002_4_00.dat 9212100 9213600
INPUT_FILE= GLA05_003_12_034_0003_1_00.dat 9213600 9215100
INPUT_FILE= GLA05_003_12_034_0003_2_00.dat 9215100 9216600
INPUT_FILE= GLA05_003_12_034_0003_3_00.dat 9216600 9218100
INPUT_FILE= GLA05_003_12_034_0003_4_00.dat 9218100 9219600
```

```
INPUT_FILE= GLA05_003_12_034_0004_1_00.dat 9219600 9221100
INPUT_FILE= GLA05_003_12_034_0004_2_00.dat 9221100 9222600
INPUT_FILE= GLA05_003_12_034_0004_3_00.dat 9222600 9224100
INPUT_FILE= GLA05_003_12_034_0004_4_00.dat 9224100 9225600
INPUT_FILE= GLA05_003_12_034_0005_1_00.dat 9225600 9227100
INPUT_FILE= GLA05_003_12_034_0005_2_00.dat 9227100 9228600
INPUT_FILE= GLA05_003_12_034_0005_3_00.dat 9228600 9230100
INPUT_FILE= GLA05_003_12_034_0005_4_00.dat 9230100 9231600
INPUT_FILE= GLA05_003_12_034_0006_1_00.dat 9231600 9233100
INPUT_FILE= GLA05_003_12_034_0006_2_00.dat 9233100 9234600
INPUT_FILE= GLA05_003_12_034_0006_3_00.dat 9234600 9236100
INPUT_FILE= GLA05_003_12_034_0006_4_00.dat 9236100 9237600
INPUT_FILE= GLA05_003_12_034_0007_1_00.dat 9237600 9239100
INPUT_FILE= GLA05_003_12_034_0007_2_00.dat 9239100 9240600
INPUT_FILE= GLA05_003_12_034_0007_3_00.dat 9240600 9242100
INPUT_FILE= GLA05_003_12_034_0007_4_00.dat 9242100 9243600
INPUT_FILE= GLA05_003_12_034_0008_1_00.dat 9243600 9245100
INPUT_FILE= GLA05_003_12_034_0008_2_00.dat 9245100 9246600
INPUT_FILE= GLA05_003_12_034_0008_3_00.dat 9246600 9248100
INPUT_FILE= GLA05_003_12_034_0008_4_00.dat 9248100 9249600
INPUT_FILE= GLA05_003_12_034_0009_1_00.dat 9249600 9251100
INPUT_FILE= GLA05_003_12_034_0009_2_00.dat 9251100 9252600
INPUT_FILE= GLA05_003_12_034_0009_3_00.dat 9252600 9254100
INPUT_FILE= GLA05_003_12_034_0009_4_00.dat 9254100 9255600
INPUT_FILE= GLA05_003_12_034_0010_1_00.dat 9255600 9257100
INPUT_FILE= GLA05_003_12_034_0010_2_00.dat 9257100 9258600
INPUT_FILE= GLA05_003_12_034_0010_3_00.dat 9258600 9260100
INPUT_FILE= GLA05_003_12_034_0010_4_00.dat 9260100 9261600
INPUT_FILE= GLA05_003_12_034_0011_1_00.dat 9261600 9263100
INPUT_FILE= GLA05_003_12_034_0011_2_00.dat 9263100 9264600
INPUT_FILE= GLA05_003_12_034_0011_3_00.dat 9264600 9266100
INPUT_FILE= GLA05_003_12_034_0011_4_00.dat 9266100 9267600
INPUT_FILE= GLA05_003_12_034_0012_1_00.dat 9267600 9269100
INPUT_FILE= GLA05_003_12_034_0012_2_00.dat 9269100 9270600
INPUT_FILE= GLA05_003_12_034_0012_3_00.dat 9270600 9272100
INPUT_FILE= GLA05_003_12_034_0012_4_00.dat 9272100 9273600
INPUT_FILE= GLA05_003_12_034_0013_1_00.dat 9273600 9275100
INPUT_FILE= GLA05_003_12_034_0013_2_00.dat 9275100 9276600
INPUT_FILE= GLA05_003_12_034_0013_3_00.dat 9276600 9278100
INPUT_FILE= GLA05_003_12_034_0013_4_00.dat 9278100 9279600
INPUT_FILE= GLA05_003_12_034_0014_1_00.dat 9279600 9281100
INPUT_FILE= GLA05_003_12_034_0014_2_00.dat 9281100 9282600
INPUT_FILE= GLA05_003_12_034_0014_3_00.dat 9282600 9284100
INPUT_FILE= GLA05_003_12_034_0014_4_00.dat 9284100 9285600

# Input GLA07 File (IF available)
#
INPUT_FILE= GLA07_003_12_034_0001_1_01.dat 9201600 9285600
#
#-----Output ANC Files
#
# Output ANC06
#
OUTPUT_FILE=anc06_003_20000417_000.dat 9201600 9285600
```

```
#  
# Output GLA06 Files  
#  
OUTPUT_FILE= GLA06_003_12_034_0001_1_00.dat 9201600 9203100  
OUTPUT_FILE= GLA06_003_12_034_0001_2_00.dat 9203100 9204600  
OUTPUT_FILE= GLA06_003_12_034_0001_3_00.dat 9204600 9206100  
OUTPUT_FILE= GLA06_003_12_034_0001_4_00.dat 9206100 9207600  
OUTPUT_FILE= GLA06_003_12_034_0002_1_00.dat 9207600 9209100  
OUTPUT_FILE= GLA06_003_12_034_0002_2_00.dat 9209100 9210600  
OUTPUT_FILE= GLA06_003_12_034_0002_3_00.dat 9210600 9212100  
OUTPUT_FILE= GLA06_003_12_034_0002_4_00.dat 9212100 9213600  
OUTPUT_FILE= GLA06_003_12_034_0003_1_00.dat 9213600 9215100  
OUTPUT_FILE= GLA06_003_12_034_0003_2_00.dat 9215100 9216600  
OUTPUT_FILE= GLA06_003_12_034_0003_3_00.dat 9216600 9218100  
OUTPUT_FILE= GLA06_003_12_034_0003_4_00.dat 9218100 9219600  
OUTPUT_FILE= GLA06_003_12_034_0004_1_00.dat 9219600 9221100  
OUTPUT_FILE= GLA06_003_12_034_0004_2_00.dat 9221100 9222600  
OUTPUT_FILE= GLA06_003_12_034_0004_3_00.dat 9222600 9224100  
OUTPUT_FILE= GLA06_003_12_034_0004_4_00.dat 9224100 9225600  
OUTPUT_FILE= GLA06_003_12_034_0005_1_00.dat 9225600 9227100  
OUTPUT_FILE= GLA06_003_12_034_0005_2_00.dat 9227100 9228600  
OUTPUT_FILE= GLA06_003_12_034_0005_3_00.dat 9228600 9230100  
OUTPUT_FILE= GLA06_003_12_034_0005_4_00.dat 9230100 9231600  
OUTPUT_FILE= GLA06_003_12_034_0006_1_00.dat 9231600 9233100  
OUTPUT_FILE= GLA06_003_12_034_0006_2_00.dat 9233100 9234600  
OUTPUT_FILE= GLA06_003_12_034_0006_3_00.dat 9234600 9236100  
OUTPUT_FILE= GLA06_003_12_034_0006_4_00.dat 9236100 9237600  
OUTPUT_FILE= GLA06_003_12_034_0007_1_00.dat 9237600 9239100  
OUTPUT_FILE= GLA06_003_12_034_0007_2_00.dat 9239100 9240600  
OUTPUT_FILE= GLA06_003_12_034_0007_3_00.dat 9240600 9242100  
OUTPUT_FILE= GLA06_003_12_034_0007_4_00.dat 9242100 9243600  
OUTPUT_FILE= GLA06_003_12_034_0008_1_00.dat 9243600 9245100  
OUTPUT_FILE= GLA06_003_12_034_0008_2_00.dat 9245100 9246600  
OUTPUT_FILE= GLA06_003_12_034_0008_3_00.dat 9246600 9248100  
OUTPUT_FILE= GLA06_003_12_034_0008_4_00.dat 9248100 9249600  
OUTPUT_FILE= GLA06_003_12_034_0009_1_00.dat 9249600 9251100  
OUTPUT_FILE= GLA06_003_12_034_0009_2_00.dat 9251100 9252600  
OUTPUT_FILE= GLA06_003_12_034_0009_3_00.dat 9252600 9254100  
OUTPUT_FILE= GLA06_003_12_034_0009_4_00.dat 9254100 9255600  
OUTPUT_FILE= GLA06_003_12_034_0010_1_00.dat 9255600 9257100  
OUTPUT_FILE= GLA06_003_12_034_0010_2_00.dat 9257100 9258600  
OUTPUT_FILE= GLA06_003_12_034_0010_3_00.dat 9258600 9260100  
OUTPUT_FILE= GLA06_003_12_034_0010_4_00.dat 9260100 9261600  
OUTPUT_FILE= GLA06_003_12_034_0011_1_00.dat 9261600 9263100  
OUTPUT_FILE= GLA06_003_12_034_0011_2_00.dat 9263100 9264600  
OUTPUT_FILE= GLA06_003_12_034_0011_3_00.dat 9264600 9266100  
OUTPUT_FILE= GLA06_003_12_034_0011_4_00.dat 9266100 9267600  
OUTPUT_FILE= GLA06_003_12_034_0012_1_00.dat 9267600 9269100  
OUTPUT_FILE= GLA06_003_12_034_0012_2_00.dat 9269100 9270600  
OUTPUT_FILE= GLA06_003_12_034_0012_3_00.dat 9270600 9272100  
OUTPUT_FILE= GLA06_003_12_034_0012_4_00.dat 9272100 9273600  
OUTPUT_FILE= GLA06_003_12_034_0013_1_00.dat 9273600 9275100  
OUTPUT_FILE= GLA06_003_12_034_0013_2_00.dat 9275100 9276600  
OUTPUT_FILE= GLA06_003_12_034_0013_3_00.dat 9276600 9278100
```

```
OUTPUT_FILE= GLA06_003_12_034_0013_4_00.dat 9278100 9279600
OUTPUT_FILE= GLA06_003_12_034_0014_1_00.dat 9279600 9281100
OUTPUT_FILE= GLA06_003_12_034_0014_2_00.dat 9281100 9282600
OUTPUT_FILE= GLA06_003_12_034_0014_3_00.dat 9282600 9284100
OUTPUT_FILE= GLA06_003_12_034_0014_4_00.dat 9284100 9285600
#
# Output GLA12-15
#
OUTPUT_FILE= GLA12_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE= GLA13_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE= GLA14_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE= GLA15_003_12_034_0001_1_00.dat 9201600 9285600
#
#-----Output QAP Files
#
# Output QAP06 Files
#
OUTPUT_FILE= QAP06_003_12_034_0001_1_00.dat 9201600 9203100
OUTPUT_FILE= QAP06_003_12_034_0001_2_00.dat 9203100 9204600
OUTPUT_FILE= QAP06_003_12_034_0001_3_00.dat 9204600 9206100
OUTPUT_FILE= QAP06_003_12_034_0001_4_00.dat 9206100 9207600
OUTPUT_FILE= QAP06_003_12_034_0002_1_00.dat 9207600 9209100
OUTPUT_FILE= QAP06_003_12_034_0002_2_00.dat 9209100 9210600
OUTPUT_FILE= QAP06_003_12_034_0002_3_00.dat 9210600 9212100
OUTPUT_FILE= QAP06_003_12_034_0002_4_00.dat 9212100 9213600
OUTPUT_FILE= QAP06_003_12_034_0003_1_00.dat 9213600 9215100
OUTPUT_FILE= QAP06_003_12_034_0003_2_00.dat 9215100 9216600
OUTPUT_FILE= QAP06_003_12_034_0003_3_00.dat 9216600 9218100
OUTPUT_FILE= QAP06_003_12_034_0003_4_00.dat 9218100 9219600
OUTPUT_FILE= QAP06_003_12_034_0004_1_00.dat 9219600 9221100
OUTPUT_FILE= QAP06_003_12_034_0004_2_00.dat 9221100 9222600
OUTPUT_FILE= QAP06_003_12_034_0004_3_00.dat 9222600 9224100
OUTPUT_FILE= QAP06_003_12_034_0004_4_00.dat 9224100 9225600
OUTPUT_FILE= QAP06_003_12_034_0005_1_00.dat 9225600 9227100
OUTPUT_FILE= QAP06_003_12_034_0005_2_00.dat 9227100 9228600
OUTPUT_FILE= QAP06_003_12_034_0005_3_00.dat 9228600 9230100
OUTPUT_FILE= QAP06_003_12_034_0005_4_00.dat 9230100 9231600
OUTPUT_FILE= QAP06_003_12_034_0006_1_00.dat 9231600 9233100
OUTPUT_FILE= QAP06_003_12_034_0006_2_00.dat 9233100 9234600
OUTPUT_FILE= QAP06_003_12_034_0006_3_00.dat 9234600 9236100
OUTPUT_FILE= QAP06_003_12_034_0006_4_00.dat 9236100 9237600
OUTPUT_FILE= QAP06_003_12_034_0007_1_00.dat 9237600 9239100
OUTPUT_FILE= QAP06_003_12_034_0007_2_00.dat 9239100 9240600
OUTPUT_FILE= QAP06_003_12_034_0007_3_00.dat 9240600 9242100
OUTPUT_FILE= QAP06_003_12_034_0007_4_00.dat 9242100 9243600
OUTPUT_FILE= QAP06_003_12_034_0008_1_00.dat 9243600 9245100
OUTPUT_FILE= QAP06_003_12_034_0008_2_00.dat 9245100 9246600
OUTPUT_FILE= QAP06_003_12_034_0008_3_00.dat 9246600 9248100
OUTPUT_FILE= QAP06_003_12_034_0008_4_00.dat 9248100 9249600
OUTPUT_FILE= QAP06_003_12_034_0009_1_00.dat 9249600 9251100
OUTPUT_FILE= QAP06_003_12_034_0009_2_00.dat 9251100 9252600
OUTPUT_FILE= QAP06_003_12_034_0009_3_00.dat 9252600 9254100
OUTPUT_FILE= QAP06_003_12_034_0009_4_00.dat 9254100 9255600
OUTPUT_FILE= QAP06_003_12_034_0010_1_00.dat 9255600 9257100
```

```
OUTPUT_FILE= QAP06_003_12_034_0010_2_00.dat 9257100 9258600
OUTPUT_FILE= QAP06_003_12_034_0010_3_00.dat 9258600 9260100
OUTPUT_FILE= QAP06_003_12_034_0010_4_00.dat 9260100 9261600
OUTPUT_FILE= QAP06_003_12_034_0011_1_00.dat 9261600 9263100
OUTPUT_FILE= QAP06_003_12_034_0011_2_00.dat 9263100 9264600
OUTPUT_FILE= QAP06_003_12_034_0011_3_00.dat 9264600 9266100
OUTPUT_FILE= QAP06_003_12_034_0011_4_00.dat 9266100 9267600
OUTPUT_FILE= QAP06_003_12_034_0012_1_00.dat 9267600 9269100
OUTPUT_FILE= QAP06_003_12_034_0012_2_00.dat 9269100 9270600
OUTPUT_FILE= QAP06_003_12_034_0012_3_00.dat 9270600 9272100
OUTPUT_FILE= QAP06_003_12_034_0012_4_00.dat 9272100 9273600
OUTPUT_FILE= QAP06_003_12_034_0013_1_00.dat 9273600 9275100
OUTPUT_FILE= QAP06_003_12_034_0013_2_00.dat 9275100 9276600
OUTPUT_FILE= QAP06_003_12_034_0013_3_00.dat 9276600 9278100
OUTPUT_FILE= QAP06_003_12_034_0013_4_00.dat 9278100 9279600
OUTPUT_FILE= QAP06_003_12_034_0014_1_00.dat 9279600 9281100
OUTPUT_FILE= QAP06_003_12_034_0014_2_00.dat 9281100 9282600
OUTPUT_FILE= QAP06_003_12_034_0014_3_00.dat 9282600 9284100
OUTPUT_FILE= QAP06_003_12_034_0014_4_00.dat 9284100 9285600
#
# Output QAP12-15
#
OUTPUT_FILE= QAP12_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE= QAP13_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE= QAP14_003_12_034_0001_1_00.dat 9201600 9285600
OUTPUT_FILE= QAP15_003_12_034_0001_1_00.dat 9201600 9285600
#
-----Execution Control
#
ELEVATION_PROCESS=ALL
#
-----End of Control File
#
=End of Control file
```

## Appendix E

# GSAS ANC06 File Contents

### E.1 ANC06 Overview

GSAS software creates ANC06 output files as required by GSMP-31100 (refer to the GLAS Science Software Requirements Document). The ANC06 file contains processing information, error messages, and status messages. The file is in standard GSAS keyword=value format. The format of an ANC06 entry is:

```
[keyword]=[value]
```

The first field [keyword] is a keyword describing the type of information presented. The second field [value] is standardized for use in ANC06. It contains two subfields: the time and a text message. The time is that of the data being processed when the entry was written (if no data have been processed, the time may be 0 or an invalid value). The time is a GSAS-standard time representation (currently UTC seconds).

```
[time] [message]
```

Each type of message may have different message field formats. The rules for parsing an ANC06 entry are:

- the keyword is the string left of the “=”
- the data time is the number immediately to the right of the “=”
- the messages text is the information after the data time

### E.2 ANC06 Example and Description

A sample ANC06 file, created during V1 Acceptance Testing, is examined below. In this example, multiple spaces have been compressed to a single space for readability.

The first entries in the example ANC06 file show what files were initially opened. The message part of each entry is a GSAS standard error/status message. The first field of the error/status message is the error code, the second is the error severity, the third is the reporting routine and the last is the error text. These fields are described in more detail in Appendix F.

```
STATUS = 0 10005, 0, OpenFInFile, Opened file: (Input) gla00_001_20000101_000001_01_00.dat
STATUS = 0 10005, 0, OpenFInFile, Opened file: (Input) anc07_001_01_00.dat
STATUS = 0 10005, 0, OpenFInFile, Opened file: (Input) anc20_001_20000101_000000_01_00.dat
STATUS = 0 10005, 0, OpenFOutFile, Opened file: (Output) gla01_001_11_001_0001_1_01_00.dat
STATUS = 0 10005, 0, OpenFOutFile, Opened file: (Output) gla02_001_11_001_0001_1_01_00.dat
STATUS = 0 10005, 0, OpenFOutFile, Opened file: (Output) gla03_001_11_001_0001_1_01_00.dat
STATUS = 0 10005, 0, OpenFInFile, Opened file: (Input) anc07_001_01_01.dat
STATUS = 0 10005, 0, OpenFInFile, Opened file: (Input) anc07_001_01_05.dat
```

The next entry is the name of the input control file.

```
CF_NAME = 0 cf01_001_20001027_001.ctl
```

The next type of entry are version numbers. Each GSAS component is separately versioned. Note that these version numbers were not updated in time for the V1 delivery and are not truly representative of the correct version of software.

```
VERSION = 0 GLAS_Exec v1_prod, 03-09-2000
VERSION = 0 libplatform v1_prod, 03-09-2000
VERSION = 0 libcntrl v1_prod, 03-09-2000
VERSION = 0 libprod vlp3, 12-09-1999
VERSION = 0 libfile vlp3, 12-09-1999
VERSION = 0 libtime vlp3, 12-09-1999
VERSION = 0 liberr v1_prod, 03-09-2000
VERSION = 0 libanc v1_prod, 03-10-2000
VERSION = 0 libl1a vlp3, 12-09-1999
VERSION = 0 libwf vlp3, 12-09-1999
VERSION = 0 libatm v1.r3, 12-09-1999
VERSION = 0 libelev vlp3, 12-09-1999
```

The next set of entries are the results of parsing the input control file. The format of the message field is exactly the same as that of the input control file with the exception of a “hit counter” added to the end. The “hit counter” displays the number of times a particular control file keyword was detected in the control file. Note that due to the length of GSAS filenames, some of the INPUT\_FILE and OUTPUT\_FILE entries are partially truncated.

```
CONTROL = 0 TEMPLATE =l1a_control_file 1
CONTROL = 0 EXEC_KEY =10023425 1
CONTROL = 0 DATE_GENERATED =2000 October 27 1
CONTROL = 0 OPERATOR =jlee 1
CONTROL = 0 CYCLE =001 0000000 0000060 1
CONTROL = 0 TRACK =001 0000000 0000060 1
CONTROL = 0 INPUT_FILE =anc07_001_01_00.dat 0000000 0000060 7
CONTROL = 0 INPUT_FILE =anc07_001_01_01.dat 0000000 0000060 7
CONTROL = 0 INPUT_FILE =anc07_001_01_05.dat 0000000 0000060 7
CONTROL = 0 INPUT_FILE =anc20_001_20000101_000000_01_00.dat 0000 7
CONTROL = 0 INPUT_FILE =gla00_001_20000101_000001_01_00.dat 0000 7
CONTROL = 0 INPUT_FILE =gla00_001_20000101_000021_01_00.dat 0000 7
CONTROL = 0 INPUT_FILE =gla00_001_20000101_000041_01_00.dat 0000 7
CONTROL = 0 OUTPUT_FILE =anc06_001_20001026_001.dat 0000000 00000 10
CONTROL = 0 OUTPUT_FILE =gla01_001_11_001_0001_1_01_00.dat 0000000 00000 10
CONTROL = 0 OUTPUT_FILE =gla01_001_11_001_0001_2_01_00.dat 0000002 10
CONTROL = 0 OUTPUT_FILE =gla01_001_11_001_0001_3_01_00.dat 0000004 10
```

```
CONTROL = 0 OUTPUT_FILE =gla02_001_11_001_0001_1_01_00.dat 000000 10
CONTROL = 0 OUTPUT_FILE =gla02_001_11_001_0001_2_01_00.dat 000002 10
CONTROL = 0 OUTPUT_FILE =gla02_001_11_001_0001_3_01_00.dat 000004 10
CONTROL = 0 OUTPUT_FILE =gla03_001_11_001_0001_1_01_00.dat 000000 10
CONTROL = 0 OUTPUT_FILE =gla03_001_11_001_0001_2_01_00.dat 000002 10
CONTROL = 0 OUTPUT_FILE =gla03_001_11_001_0001_3_01_00.dat 000004 10
```

Notice that for the execution control keywords, the input keywords may not be shown exactly as set in the control file. In the following case, the control file had L1A\_PROCESS=ALL in it. What is shown below is that the control file parser sets flags based on the input keywords. The ANC06 file shows the resultant value of the affected flags. The control file section gives allowable values for the execution keywords.

```
CONTROL = 0 L1A_PROCESS = L_Alt 1
CONTROL = 0 L1A_PROCESS = L_AtM 1
CONTROL = 0 L1A_PROCESS = L_Att 1
CONTROL = 0 L1A_PROCESS = L_Eng 1
CONTROL = 0 WAVEFORM_OPTION = W_doAllRgns 0
CONTROL = 0 WAVEFORM_OPTION = W_processLand 0
CONTROL = 0 WAVEFORM_OPTION = W_processOcean 0
CONTROL = 0 WAVEFORM_OPTION = W_processIceS 0
CONTROL = 0 WAVEFORM_OPTION = W_processSeaI 0
```

The next set of entries show that the static ancillary files which were read into core have been closed.

```
STATUS = 0 10006, 0, CloseFile, Closed file: anc07_001_01_00.dat
STATUS = 0 10006, 0, CloseFile, Closed file: anc07_001_01_01.dat
STATUS = 0 10006, 0, CloseFile, Closed file: anc07_001_01_05.dat
```

The next set of entries show the version numbers of the static ancillary files which were read into core. If a particular static ANC file was not read, the version number is replaced with the appropriate information.

```
VERSION = 0 ANC07 Errors V1 07 November 2000
VERSION = 0 ANC07 Status V1 07 November 2000
VERSION = 0 ANC07 Globals v1.1 13 March 2000
VERSION = 0 no ANC07 Atm globals read
VERSION = 0 ANC07 l1a Globals v1.2 Wed Mar 8 09:10:
VERSION = 0 no ANC07 WF globals read
VERSION = 0 no ANC07 Elev globals read
```

The next set of entries show a progression of status messages tracing process flow. Note how time progressively increments.

```
STATUS = 1 50006, 0, C_IntrpPOD, IN C_IntrpPOD subroutine status
STATUS = 1 50008, 0, C_CalcSpLoc, IN C_CALCSPLOC subroutine status
```

```

STATUS = 1 20003, 0, L_Eng_Proc, L_Eng subroutine status Entering subroutine
STATUS = 1 20001, 0, L_Alt_Proc, L_Alt subroutine status Entering subroutine
STATUS = 1 20002, 0, L_Atm_Proc, L_Atm subroutine status Entering subroutine
STATUS = 10 20003, 0, L_Eng_Proc, L_Eng subroutine status Entering subroutine
STATUS = 10 20001, 0, L_Alt_Proc, L_Alt subroutine status Entering subroutine
STATUS = 10 20002, 0, L_Atm_Proc, L_Atm subroutine status Entering subroutine
STATUS = 20 20003, 0, L_Eng_Proc, L_Eng subroutine status Entering subroutine
STATUS = 20 20001, 0, L_Alt_Proc, L_Alt subroutine status Entering subroutine
STATUS = 20 20002, 0, L_Atm_Proc, L_Atm subroutine status Entering subroutine

```

**The next set of entries show that a granule has ended and is closed and that a new granule has been opened.**

```

STATUS = 2147483647 10007, 0, NextGranule, Finished reading file:
gla00_001_20000101_000001_01_00.dat

STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla00_001_20000101_000001_01_00.dat
STATUS = 2147483647 10005, 0, OpenFInFile, Opened file: (Input)
gla00_001_20000101_000021_01_00.dat

STATUS = 21 10006, 0, CloseFile, Closed file: gla01_001_11_001_0001_1_01_00.dat
STATUS = 21 10005, 0, OpenFOutFile, Opened file: (Output)
gla01_001_11_001_0001_2_01_00.dat

STATUS = 21 10006, 0, CloseFile, Closed file: gla02_001_11_001_0001_1_01_00.dat
STATUS = 21 10005, 0, OpenFOutFile, Opened file: (Output)
gla02_001_11_001_0001_2_01_00.dat

STATUS = 21 10006, 0, CloseFile, Closed file: gla03_001_11_001_0001_1_01_00.dat
STATUS = 21 10005, 0, OpenFOutFile, Opened file: (Output)
gla03_001_11_001_0001_2_01_00.dat

```

**Progress of the processing is again tracked, including more granule switches...**

```

STATUS = 25 50006, 0, C_IntrpPOD, IN C_IntrpPOD subroutine status
STATUS = 25 50008, 0, C_CalcSpLoc, IN C_CALCSPLOC subroutine status

STATUS = 30 20003, 0, L_Eng_Proc, L_Eng subroutine status Entering subroutine
STATUS = 30 20001, 0, L_Alt_Proc, L_Alt subroutine status Entering subroutine
STATUS = 30 20002, 0, L_Atm_Proc, L_Atm subroutine status Entering subroutine
STATUS = 40 20003, 0, L_Eng_Proc, L_Eng subroutine status Entering subroutine
STATUS = 40 20001, 0, L_Alt_Proc, L_Alt subroutine status Entering subroutine
STATUS = 40 20002, 0, L_Atm_Proc, L_Atm subroutine status Entering subroutine
STATUS = 2147483647 10007, 0, NextGranule, Finished reading file:
gla00_001_20000101_000021_01_00.dat

STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla00_001_20000101_000021_01_00.dat
STATUS = 2147483647 10005, 0, OpenFInFile, Opened file: (Input)
gla00_001_20000101_000041_01_00.dat

STATUS = 41 10006, 0, CloseFile, Closed file: gla01_001_11_001_0001_2_01_00.dat
STATUS = 41 10005, 0, OpenFOutFile, Opened file: (Output)
gla01_001_11_001_0001_3_01_00.dat

```

---

```

STATUS = 41 10006, 0, CloseFile, Closed file: gla02_001_11_001_0001_2_01_00.dat
STATUS = 41 10005, 0, OpenFOutFile, Opened file: (Output)
gla02_001_11_001_0001_3_01_00.dat

STATUS = 41 10006, 0, CloseFile, Closed file: gla03_001_11_001_0001_2_01_00.dat
STATUS = 41 10005, 0, OpenFOutFile, Opened file: (Output)
gla03_001_11_001_0001_3_01_00.dat

STATUS = 50 50006, 0, C_IntrpPOD, IN C_IntrpPOD subroutine status
STATUS = 50 50008, 0, C_CalcSpLoc, IN C_CALCSPLOC subroutine status
STATUS = 50 20003, 0, L_Eng_Proc, L_Eng subroutine status Entering subroutine
STATUS = 50 20001, 0, L_Alt_Proc, L_Alt subroutine status Entering subroutine
STATUS = 50 20002, 0, L_AtM_Proc, L_AtM subroutine status Entering subroutine
STATUS = 60 20003, 0, L_Eng_Proc, L_Eng subroutine status Entering subroutine
STATUS = 60 20001, 0, L_Alt_Proc, L_Alt subroutine status Entering subroutine
STATUS = 60 20002, 0, L_AtM_Proc, L_AtM subroutine status Entering subroutine
STATUS = 2147483647 10007, 0, NextGranule, Finished reading file:
gla00_001_20000101_000041_01_00.dat

STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla00_001_20000101_000041_01_00.dat

```

**At the end of processing, a summary section lists the number of records read or written to/from each file. Note that since processing is complete and there is no valid time of data, the time is invalid.**

```

INPUT_SUMMARY = 2147483647 gla00_001_20000101_000001_01_00.dat: read 19 records
INPUT_SUMMARY = 2147483647 gla00_001_20000101_000021_01_00.dat: read 19 records
INPUT_SUMMARY = 2147483647 gla00_001_20000101_000041_01_00.dat: read 19 records
INPUT_SUMMARY = 2147483647 anc20_001_20000101_000000_01_00.dat: read 0 records
OUTPUT_SUMMARY = 2147483647 gla01_001_11_001_0001_1_01_00.dat: wrote 90 records
OUTPUT_SUMMARY = 2147483647 gla01_001_11_001_0001_2_01_00.dat: wrote 90 records
OUTPUT_SUMMARY = 2147483647 gla01_001_11_001_0001_3_01_00.dat: wrote 90 records
OUTPUT_SUMMARY = 2147483647 gla02_001_11_001_0001_1_01_00.dat: wrote 20 records
OUTPUT_SUMMARY = 2147483647 gla02_001_11_001_0001_2_01_00.dat: wrote 20 records
OUTPUT_SUMMARY = 2147483647 gla02_001_11_001_0001_3_01_00.dat: wrote 20 records
OUTPUT_SUMMARY = 2147483647 gla03_001_11_001_0001_1_01_00.dat: wrote 1 records
OUTPUT_SUMMARY = 2147483647 gla03_001_11_001_0001_2_01_00.dat: wrote 1 records
OUTPUT_SUMMARY = 2147483647 gla03_001_11_001_0001_3_01_00.dat: wrote 1 records

```

**The second part of the summary section lists all error/status messages detected during the processing. The first field of the message is the number of times that particular error/status code was detected.**

```

STATUS_SUMMARY = 2147483647 8 Opened file:
STATUS_SUMMARY = 2147483647 12 Closed file:
STATUS_SUMMARY = 2147483647 3 Finished reading file:
STATUS_SUMMARY = 2147483647 60 L_Alt subroutine status
STATUS_SUMMARY = 2147483647 60 L_AtM subroutine status

```

```
STATUS_SUMMARY = 2147483647 60 L_Eng subroutine status
STATUS_SUMMARY = 2147483647 2400 IN C_IntrpPOD subroutine status
STATUS_SUMMARY = 2147483647 2400 IN C_CALCSPLLOC subroutine status
```

**The final section of the ANC06 file shows that any opened files are closed.**

```
STATUS = 2147483647 10006, 0, CloseFile, Closed file: anc20_001_20000101_000000_01_00.dat
STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla01_001_11_001_0001_1_01_00.dat
STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla01_001_11_001_0001_2_01_00.dat
STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla01_001_11_001_0001_3_01_00.dat
STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla02_001_11_001_0001_1_01_00.dat
STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla02_001_11_001_0001_2_01_00.dat
STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla02_001_11_001_0001_3_01_00.dat
STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla03_001_11_001_0001_1_01_00.dat
STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla03_001_11_001_0001_2_01_00.dat
STATUS = 2147483647 10006, 0, CloseFile, Closed file: gla03_001_11_001_0001_3_01_00.dat
```

## Appendix F

# GSAS Error and Status Codes

### F.1 Invalid Values

Not all data received from GLAS will be suitable for science processing. In addition, given the nature of the raw telemetry packets, some data may be missing. The concept of an “invalid value” is used to signify that data is invalid or missing and should not be used for processing. Invalid values are datatype-specific values which are defined in the GLAS global constants module. These variables are assigned to Product File variables in order to indicate invalid or missing data. The current values are defined in Table F-1.

**Table F-1 Invalid Values**

Datatype	Invalid Value
1 byte integer	127
2 byte integer	32767
4 byte integer	2147483647
4 byte real	3.40282E+38 x7FFFFFFF
8 byte real	1.797693094862316E+308 x7FFFFFFFFFFFFF

### F.2 Error/Status Format

All GSAS use a common error/status reporting technique. Errors are read from an standard GLAS “keyword=value” ANC07 file. The two fields are a keyword and an value (error string). The format is defined in Figure F-1. The KEYWORD can have the

KEYWORD=nnnnnnxttxsxfffff

**Figure F-1 Error Ancillary File Format**

value of “ERROR” or “STATUS” and identifies if the line contains an error or status entry. The VALUE is a text string with the specific format defined in Table F-2.

**Table F-2 Error String Format**

Character	Positions	Description
n	1-6	Error code (must be sequential within a section)
x	7,58,60	Space character (delimiter)

**Table F-2 Error String Format (Continued)**

Character	Positions	Description
t	8-57	Message
s	59	Error severity (see Table F-3)
f	61-66	Frequency of reporting (message is reported on 1st occurrence, then every f'th time)

There is a specific error number for each error/status value. Within the ANC07 file, these error numbers are numerically split into multiple sub-sections. Errors have negative numeric designations; status messages have positive designations.

Each major portion of the GSAS software supported by the specific error file begins at a different subsection number. Within a subsection, error numbers must be consecutive. The use of sub-sectioning is optional for a simple error file. The GLAS\_Exec error file, for example, contains a separate subsection for each subsystem. Sections are defined to start at the following error numbers: 0, 10000, 20000, 30000, 40000, 50000, 60000, 70000, 80000 and 90000.

### F.3 Error/Status Usage

GLAS error messages are designed to inform a user when the software has encountered a problem. GLAS status messages are designed to assist the user in observing the flow of the processing. Status messages usually alert the user when the software begins execution of a subroutine. A great deal of flexibility was designed into this software in order to allow the user to customize the error/status display.

The user may modify error and status entries in order to configure the severity of the error and frequency of printout. The user is cautioned to seek GLAS change-control board approval before modifying the severity of an error. GSAS software will terminate processing upon receipt of a fatal severity code. Thus, modifying the severity may enable the software to execute in a non-tested mode.

The severity column control how the GLAS software reacts when an error occurs. The 4 levels of severity are described in Table F-3. GLAS software will terminate on a Fatal error. The frequency column controls how often an error message is printed out. The first instance of a specific error is always printed. Subsequent instances are printed out at the frequency specified. All instances are counted and the number of occurrences printed in the output summary. .

**Table F-3 Error Severity Codes**

Severity	Description
0	No error
1	Information/status

**Table F-3 Error Severity Codes (Continued)**

Severity	Description
2	Warning
3	Fatal

The GSAS ANC07 error file has 5 subsections. Table F-4 lists each of the subsections and their starting error number.

**Table F-4 GLAS\_Exec Error Sections**

Starting Error Number	Description
-10001	General errors.
-20001	L1A errors.
-30001	Waveform errors
-40001	Atmosphere errors
-50001	Elevation errors.

## F.4 GSAS Error Codes

This section list errors and possible solutions for each error defined in GSAS.

### F.4.1 General Errors

#### **-10001 Error Opening File for Input.**

The operating system was unable to open the specified input file. Verify that the requested file exists in (or is correctly linked into) the current directory. Verify the spelling of the filename within the control file (filenames are case-sensitive). Verify that the user has read permission on the file.

#### **-10002 No control file specified.**

A control file must be specified as a command-line argument when GLAS\_Exec is executed. If a control file was specified, then perform the following verification steps:  
Verify that the control file exists in (or is correctly linked into) the current directory.  
Verify that the user has read permission on the control file

#### **-10003 Bad rec\_ndx in control file.**

A bad start or stop time was specified in the control file. The start/stop times are fractional numbers representing UTC seconds. The times must contain only digits and the decimal point.

#### **-10004 Stop rec\_ndx > start rec\_ndx.**

A start time which was chronologically greater than the stop time was found in the control file. Start times must be less than stop times.

#### **-10005 Previous granule stop > current start.**

Multiple granules of the same type were specified in the control file. The start time of

a previous granule was chronologically greater than the stop time of the current granules. Multiple granules of the same type must not have time spans which overlap within the same control file. The granules must be entered in the control file in chronological order.

**-10006 Error Opening File for Output:**

The operating system was unable to open the specified output file. Verify that the requested file does not already exists in (or is correctly linked into) the current directory. Verify that the user has write permission on the current directory.

**-10007 Error Closing File:**

The operating system returned an error code when attempting to close a file. Verify that opened files were not moved from the current directory while GLAS\_Exec was running. This is an unusual error and would normally indicate a programming error.

**-10008 Error Reading File:**

The operating system returned an error code when attempting to read a file. Verify that files listed in the control file are present and accessible in the current directory. This is an unusual error and would normally indicate a programming error or a corrupt input file.

**-10009 Error Writing File:**

The operating system returned an error code when attempting to write a file. Verify that files listed in the control file are present and accessible in the current directory. This is an unusual error and would normally indicate a programming error or a full file system.

**-10010 Error Reading ANC07:**

The operating system returned an error code when attempting to read the specified ANC07 file. Verify that the ANC07 files listed in the control file are present and accessible in the current directory. If all files are present, this indicates corruption in the ANC07 file. Appendix B describes the format of ANC07 files. Verify the ANC07 file contents.

**-10011 Unknown Keyword in ANC07:**

The ANC07 file reader has found a keyword it does not recognize in the ANC07 file. This usually indicates corruption in the ANC07 file. Appendix B lists recognized ANC07 keywords. Verify the ANC07 file contents.

**-10012 Multiple single-instance keywords:**

The control file parser has detected multiple instances of single-instance keywords in the control file. Appendix C details single and multiple-instance keywords in the control file. Verify the control file contents.

**-10013 Multiple-instance keyword limit exceeded:**

The control file parser has detected that the number of instances of a multiple-instance keyword has been exceeded. Appendix C details single and multiple-instance keywords in the control file. Verify the control file contents.

**-10014 Unrecognized line in control file:**

The control file parser has found a line in the control file which it does not recognize. Appendix C details the format of the control file. Verify the control file contents.

**-10015 Unknown value in keyword/value pair:**

The control file parser has found a line in the control file with a value it does not recognize. Certain keywords are required to have only certain values. Appendix C lists possible values within the control file. Verify the control file contents.

**-10016 I/O Error Opening Control File:**

The operating system was unable to open the specified control file. Verify that the requested file exists in (or is correctly linked into) the current directory. Verify the spelling of the filename on the command line. Verify that the user has read permission on the file.

**-10017 Error Reading Control File:**

The operating system returned an error code when attempting to read the control file. Verify that the control file is present and accessible in the current directory. This is an unusual error and would normally indicate a programming error or a corrupt control file.

**-10018 Specified Unknown File Type:**

An unknown file type was specified in the control file. A file type is represented by the first 5 characters of the filename. Appendix A lists recognized GLAS file types. Verify the control file.

**-10019 GLA01 Unknown Record Type**

An unknown record type was found within the GLA01 file. Valid record types are header, body, long and short. This error indicates a corrupt GLA01 file or a program error.

**-10020 GLA01 Exceeded Waveform Record Limit**

The number of waveform records per second was exceeded in the GLA01 file. This error indicates a corrupt GLA01 file or a program error.

**-10021 GLA00 Unknown APid**

An unknown APID was detected in the GLA00 data. This error indicates corrupt GLA00 data.

**-10022 GLA00 Wrong APid**

This error is not used in GLAS\_Exec V1.

**-10023 Max APIDs per sec exceeded**

This indicates that the maximum number of APIDs per second has been exceeded. This error indicates corrupt GLA00 data.

**-10024 Data Time < Granule Start Time**

This indicates that data was lost because the time of the current record is less than the start time of the current granule. Verify the start and stop times in the control file.

**-10025 Error Reading Standard Atmosphere file header**

An error was detected reading the Standard Atmosphere file header. This indicates a corrupt Standard Atmosphere file.

**-10026 Error Reading Standard Atmosphere file data**

An error was detected reading the Standard Atmosphere file data. This indicates a corrupt Standard Atmosphere file.

**-10027 Error Reading Global Aerosol map**

An error was detected reading the Global Aerosol file data. This indicates a corrupt Global Aerosol file.

**-10028 Error Reading Aerosol Troposphere map**

An error was detected reading the Aerosol troposphere file data. This indicates a corrupt Aerosol Troposphere map file.

**F.4.2 L1A Errors****-20001 Error reading PAD data Eng data**

This error is not used in GLAS\_Exec V1.

**-20002 Error reading PAD data Eng data**

This error is not used in GLAS\_Exec V1.

**F.4.3 Waveform Errors**

Subsystem errors and corrective action will be fully documented in V2 of this document.

**-30001 Singular Matrix****-30002 Number of peaks found greater than max****-30003 No signal****-30004 Standard deviation is zero****-30005 Bad Frame****-30006 Error from C\_InterpPOD****-30007 Error from C\_CalcSpLoc****-30008 Error from WF\_Mgr****F.4.4 Atmosphere Errors**

Subsystem errors and corrective action will be fully documented in V2 of this document.

**-40001 DEM out-of-bounds****-40002 532 integrated return flag poor or bad****-40003 Ratio of integrated returns out-of-bounds****-40004 Large num bad recs not incl in data**

- 40005 Excessive num bad recs not incl in data**
- 40006 Time between recs greater than threshold**
- 40007 Divide by zero**
- 40008 Exponent too large**
- 40009 Index beyond gi\_cld\_lays**
- 40010 Index beyond gi\_pb\_aer\_lays**
- 40011 Deficient 532 laser energy flag**
- 40012 Deficient 1064 laser energy flag**
- 40013 Excess num cloudy recs not incl in data**
- 40014 PBL signal ratio less than zero**
- 40015 Input PBL height out-of-bounds**
- 40016 Aerosol signal ratio less than zero**
- 40017 Insufficient filtered data for processing**
- 40018 Random noise not valid for cloud detection**
- 40019 Too many bad locations**
- 40020 PBL QA array index beyond array size**
- 40021 Invalid lat/lon from GLA07**

#### F.4.5 Elevation Errors

Subsystem errors and corrective action will be fully documented in V2 of this document.

##### **-50001 Error reading met height file**

An error was detected reading the met height file. This indicates a corrupt met height file.

##### **-50002 Error reading met temperature file**

An error was detected reading the met temperature file. This indicates a corrupt met temperature file.

##### **-50003 Error reading met relative humidity file**

An error was detected reading the met relative humidity file. This indicates a corrupt met relative humidity file.

##### **-50004 Error reading met precip water table file**

An error was detected reading the met precip water file. This indicates a corrupt met precip water file.

##### **-50005 Elevation more than max pressure level ht**

##### **-50006 Max Iterations exceeded in ODE calcs.**

**-50007 Step size underflow in BS-step method**

**-50008 Value of cosine GT 1.0 in Geoloc**

**-50009 Value of cosine LT -1.0 in Geoloc**

**-50010 Max Iterations exceeded in Geoloc**

**-50011 POD file structure is empty**

**-50012 Time not within vals in POD file structure**

**-50013 Error reading 10 consec recs in POD file**

**-50014 Error reading Load Tide coeffs file**

An error was detected reading the load tide coefficients file. This indicates a corrupt load tide coefficients file.

**-50015 Error reading Geoid coeffs file**

An error was detected reading the geoid file. This indicates a corrupt geoid file.

**-50016 Error reading DEM file**

An error was detected reading the DEM file. This indicates a corrupt DEM file.

**-50017 Error reading 1 deg Ocean Mask file**

An error was detected reading the ocean mask file. This indicates a corrupt ocean mask file.

**-50018 Size of ocean td array must be increased**

**-50019 Likely error in Check1 of legacy ocean td**

**-50020 Lat/Lon out of bounds in legacy ocean td**

**-50021 Increase dimensions in call to UTCSRI**

**-50022 Bad Frame flag set in E\_IceSheetParm(rem)**

**-50023 Bad Frame flag set in E\_SeaIceParm (rem)**

**-50024 Bad Frame flag set in E\_LandParm (rem)**

**-50025 Bad Frame flag set in E\_OceanParm (rem)**

**-50026 Bad Frame flag set in E\_CalcSlope (rem)**

**-50027 Bad Frame flag set in E\_CalcRange (rem)**

**-50028 Bad Frame flag set in E\_CalcRngOff (rem)**

**-50029 ROT file structure is empty**

**-50030 Time not within vals in ROT file structure**

**-50031 Error reading recs in ROT file**

An error was detected reading the rotation file. This indicates a corrupt rotation file.

## F.5 HP Runtime Error Codes

This section provides a list of HP runtime error codes. These codes are taken directly from *HP Fortran 90 Programmers Reference; Product Number: B3909DB; Fortran 90 Compiler for HP-UX; Document Number: B3908-90002; October 1998.*

Error No.	Error Message	Description	Action
900	ERROR IN FORMAT	FORMAT statement syntax contains an error.	Refer to the syntax for "FORMAT" on page 330. Also see Chapter 8, "I/O and file handling," on page 171 for the syntax of the format specification and edit descriptors.
901	NEGATIVE UNIT NUMBER SPECIFIED	Unit number was not greater than or equal to zero.	Use a non-negative unit number.
902	FORMATTED I/O ATTEMPTED ON UNFORMATTED FILE	Formatted I/O was attempted on a file opened for unformatted I/O.	Open the file for formatted I/O or perform unformatted I/O on this file. <sup>9</sup>
903	UNFORMATTED I/O ATTEMPTED ON FORMATTED FILE	Unformatted I/O was attempted on a file opened for formatted I/O.	Open the file for unformatted I/O or perform formatted I/O on this file.
904	DIRECT I/O ATTEMPTED ON SEQUENTIAL FILE	Direct operation attempted on sequential file, direct operation attempted on opened file connected to a terminal.	Use sequential operations on this file, open file for direct access, or do not do direct I/O on a file connected to a terminal.
905	ERROR IN LIST-DIRECTED READ OF LOGICAL DATA	Found repeat value, but no asterisk. First character after optional decimal point was not T or F.	Change input data to correspond to syntax expected by list-directed input of logicals, or use input statement that corresponds to syntax of input data.
907	ERROR IN LIST-DIRECTED READ OF CHARACTER DATA	Found repeat value, but no asterisk. Characters not delimited by quotation marks.	Change input data to correspond to syntax expected by list-directed input of characters, or use input statement that corresponds to syntax of input data.

Error No.	Error Message	Description	Action
908	COULD NOT OPEN FILE SPECIFIED	Tried to open a file that the system would not allow for one of the following reasons: access to the file was denied by the file system due to access restriction; the named file does not exist; or the type of access request is impossible.	Correct the pathname to open the intended file.
909	SEQUENTIAL I/O ATTEMPTED ON DIRECT ACCESS FILE	Attempted a BACKSPACE, REWIND, or ENDFILE on a terminal or other device for which these operations are not defined.	Do not use the BACKSPACE, REWIND, and ENDFILE statements.
910	ACCESS PAST END OF RECORD ATTEMPTED	Tried to do I/O on record of a file past beginning or end of record.	Perform I/O operation within bounds of the record, or increase record length.
912	ERROR IN LIST I/O READ OF COMPLEX DATA	While reading complex data, one of the following problems has occurred: no left parenthesis and no repeat value; repeat value was found but no asterisk; or no closing right parenthesis.	Change input data to correspond to syntax expected by list-directed input of complex numbers, or use input statement corresponding to syntax of input data.
913	OUT OF FREE SPACE	Library cannot allocate an I/O block (from an OPEN statement), parse array (for formats assembled at run-time), file name string (from OPEN) characters from list-directed read, or file buffer. The program may be trying to overwrite a shared memory segment defined by another process.	Allocate more free space in the heap area, open fewer files, use FORMAT statements in place of assembling formats at run time in character arrays, or reduce the maximum size of file records.
914	ACCESS OF UNCONNECTED UNIT ATTEMPTED	Unit specified in I/O statement has not previously been connected to anything.	Connect unit using the OPEN statement before attempting I/O on it, or perform I/O on another, already connected, unit.
915	READ UNEXPECTED CHARACTER	Read a character that is not admissible for the type of conversion being performed. Input value was too large for the type of the variable.	Remove from input data any characters that are illegal in integers or real numbers.

Error No.	Error Message	Description	Action
916	ERROR IN READ OF LOGICAL DATA	An illegal character was read when logical data was expected.	Change input data to correspond to syntax expected when reading logical data or use input statement corresponding to syntax of input data.
917	OPEN WITH NAMED SCRATCH FILE ATTEMPTED	Executed OPEN statement with STATUS='SCRATCH', but also named the file. Scratch files must not be named.	Either remove the FILE= specifier, or open the file with a status other than STATUS='SCRATCH'.
918	OPEN OF EXISTING FILE WITH STATUS='NEW' ATTEMPTED	Executed OPEN statement with STATUS='NEW', but file already exists.	Either remove the STATUS= specifier from the OPEN statement, or use the STATUS='OLD'; STATUS='UNKNOWN'; or STATUS='REPLACE' specifier.
920	OPEN OF FILE CONNECTED TO DIFFERENT UNIT ATTEMPTED	You attempted to open a file that is already open with a different unit number.	Close the file with the current unit number before attempting to open it with a different unit number.
921	UNFORMATTED OPEN WITH BLANK SPECIFIER ATTEMPTED	OPEN statement specified FORM='UNFORMATTE D' and BLANK=xx.	Either use FORM='FORMATTED' or remove BLANK=xx.
922	READ ON ILLEGAL RECORD ATTEMPTED	Attempted to read a record of a formatted or unformatted direct file that is beyond the current end-of-file.	Read records that are within the bounds of the file.
923	OPEN WITH ILLEGAL FORM SPECIFIER ATTEMPTED	FORM= specified string other than 'FORMATTED' or 'UNFORMATTED'.	Use either 'FORMATTED' or 'UNFORMATTED' for the FORM= specifier in an OPEN statement.
924	CLOSE OF SCRATCH FILE WITH STATUS='KEEP' ATTEMPTED	The file specified in the CLOSE statement was previously opened with 'SCRATCH' specified in the STATUS= specifier.	Open the file with a STATUS=, specifying a string other than 'SCRATCH' or do not specify STATUS='KEEP' in the CLOSE statement for this scratch file.
925	OPEN WITH ILLEGAL STATUS SPECIFIER ATTEMPTED	STATUS= specified string other than 'OLD' 'NEW' 'UNKNOWN' 'REPLACE' or 'SCRATCH'.	Use 'OLD', 'NEW', 'UNKNOWN', 'REPLACE' or 'SCRATCH' for the STATUS= specifier in OPEN statement.

Error No.	Error Message	Description	Action
926	CLOSE WITH ILLE-GAL STATUS SPECIFIER ATTEMPTED	STATUS= specified string other than 'KEEP' or 'DELETE'.	Use 'KEEP' or 'DELETE' for the STATUS= specifier in a CLOSE statement.
927	OPEN WITH ILLE-GAL ACCESS SPECIFIER ATTEMPTED	ACCESS= specified string other than 'SEQUENTIAL' or 'DIRECT'.	Use 'SEQUENTIAL' or 'DIRECT' for the ACCESS= specifier in an OPEN statement.
929	OPEN OF DIRECT FILE WITH NO RECL SPECIFIER ATTEMPTED	OPEN statement has ACCESS='DIRECT', but no RECL= specifier.	Add RECL= specifier to OPEN statement, or specify ACCESS= 'SEQUENTIAL'.
930	OPEN WITH RECL LESS THAN 1 ATTEMPTED	RECL= specifier in OPEN statement was less than or equal to zero.	Specify a positive number for RECL= specifier in OPEN statement.
931	OPEN WITH ILLE-GAL BLANK SPECIFIER ATTEMPTED	BLANK= specified string other than 'NULL' or 'ZERO'	Use 'NULL' or 'ZERO' for BLANK= specifier in OPEN statement.
933	END (OR BEGIN) OF FILE WITH NO END=x SPECIFIER	End-of-file mark read by a READ statement with no END= specifier to indicate label to which to jump.	Use the END= specifier to handle EOF, or check logic.
937	ILLEGAL RECORD NUMBER SPECIFIED	A record number less than one was specified for direct I/O.	Use record numbers greater than zero.
942	ERROR IN LIST-DIRECTED READ - CHARACTER DATA READ FOR ASSIGNMENT TO NON-CHARACTER VARIABLE	A character string was read for a numerical or logical variable.	Check input data and input variable type.
944	RECORD TOO LONG IN DIRECT UNFORMATTED I/O	Output requested is too long for specified (or pre-existing) record length.	Make the number of bytes output by WRITE less than or equal to the file record size.
945	ERROR IN FORMATTED I/O	More bytes of I/O were requested than exist in the current record.	Match the format to the data record.
953	NO REPEATABLE EDIT DESCRIPTOR IN FORMAT STRING	No format descriptor was found to match I/O list items.	Add at least one repeatable edit descriptor to the format statement.
956	FILE SYSTEM ERROR	The file system returned an error status during an I/O operation.	See the associated file system error message.

Error No.	Error Message	Description	Action
957	FORMAT DESCRIPTOR INCOMPATIBLE WITH NUMERIC ITEM IN I/O LIST	A numeric item in the I/O list was matched with a non-numeric edit descriptor.	Match format descriptors to I/O list.
958	FORMAT DESCRIPTOR INCOMPATIBLE WITH CHARACTER ITEM IN I/O LIST	A character item in the I/O list was matched with an edit descriptor other than A or R.	Match format descriptors to I/O list.
959	FORMAT DESCRIPTOR INCOMPATIBLE WITH LOGICAL ITEM IN I/O LIST	A logical item in the I/O list was matched with a edit descriptor other than L.	Match format descriptors to I/O list.
973	RECORD LENGTH DIFFERENT IN SUBSEQUENT OPEN	Record length specified in second OPEN conflicted with the value as opened.	Only BLANK=, DELIM=, and PAD= specifiers may be changed by a redundant OPEN.
974	RECORD ACCESSED PAST END OF INTERNAL FILE RECORD (VARIABLE)	An attempt was made to transfer more characters than internal file length.	Match READ or WRITE statement with internal file size.
975	ILLEGAL NEW FILE NUMBER REQUESTED IN FSET FUNCTION	The file number requested to be set was not a legal file system file number.	Check that the OPEN succeeded and the file number is correct.
976	UNEXPECTED CHARACTER IN "NAMELIST" READ	An illegal character was found in namelist-directed input.	Be sure input data conforms to the syntax rules for namelist-directed input, or remove illegal character from data.
977	ILLEGAL SUBSCRIPT OR SUBSTRING IN "NAMELIST" READ	An invalid subscript or substring specifier was found in namelist-directed input. Possible causes include bad syntax, subscript/substring component out-of-bounds, wrong number of subscripts and substring on non-character variable.	Check input data for syntax errors. Be sure subscript/substring specifiers are correct for data type. Specify only array elements within the bounds of the array being read.
978	TOO MANY VALUES IN "NAMELIST" READ	Too many input values were found during a namelist-directed READ. This message will be generated by attempts to fill variables beyond their memory limits.	Supply only as many values as the length of the array.

Error No.	Error Message	Description	Action
979	VARIABLE NOT IN NAMELIST GROUP	A variable name was encountered in the input stream that was not declared as part of the current namelist group.	Read only the variables in this namelist.
980	NAMELIST I/O ATTEMPTED ON UNFORMATTED FILE	An illegal namelist-directed I/O operation was attempted on an unformatted (binary) file.	Specify FORM='FORMATTED' in OPEN statement, or use namelist-directed I/O only on formatted files.
1010	OPEN WITH ILLEGAL PAD SPECIFIER ATTEMPTED	An attempt was made to open a file with an illegal value specified for the PAD= specifier.	Specify either PAD='YES' or PAD='NO'.
1011	OPEN WITH ILLEGAL POSITION SPECIFIER ATTEMPTED	An attempt was made to open a file with an illegal value specified for the POSITION= specifier.	Specify POSITION='ASIS', POSITION='REWIND' or POSITION='APPEND'.
1012	OPEN WITH ILLEGAL DELIM SPECIFIER ATTEMPTED	An attempt was made to open a file with an illegal value specified for the DELIM= specifier.	Specify DELIM= 'APOSTROPHE', DELIM='QUOTE' or DELIM='NONE'.
1013	OPEN WITH ILLEGAL ACTION SPECIFIER ATTEMPTED	An attempt was made to open a file with an illegal value specified for the ACTION= specifier.	Specify ACTION='READ', ACTION='WRITE' or ACTION='READWRITE'.

## Abbreviations & Acronyms

ALT	designation for the EOS-Altimeter spacecraft series
DAAC	Distributed Active Archive Center
EDOS	EOS Data and Operations System
EOC	EOS Operating Center
EOS	NASA Earth Observing System Mission Program
EOSDIS	Earth Observing System Data and Information System
GDS	GLAS Ground Data System
GLAS	Geoscience Laser Altimeter System instrument or investigation
GPS	Global Positioning System
GSFC	NASA Goddard Space Flight Center at Greenbelt, Maryland
GSFC/WFF	NASA Goddard Space Flight Center/Wallops Flight Facility at Wallops Island, Virginia
ID	Identification
IEEE	Institute for Electronics and Electrical Engineering
IST	GLAS Instrument Support Terminal
LASER	Light Amplification by Stimulated Emission of Radiation
LIDAR	Light Detection and Ranging
N/A	Not (/) Applicable
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
POD	Precision Orbit Determination
SCF	GLAS investigation Science Computing Facility and workstation(s)
SDPS	Science Data Processing Segment
TBD	to be determined, to be done, or to be developed
UNIX	the operating system jointly developed by the AT&T Bell Laboratories and the University of California-Berkeley System Division



# Glossary

aggregate	A collection, assemblage, or grouping of distinct data parts together to make a whole. It is generally used to indicate the grouping of GLAS data items, arrays, elements, and EOS parameters into a data record. For example, the collection of Level 1B EOS Data Parameters gathered to form a one-second Level 1B data record. It could be used to represent groupings of various GLAS data entities such as data items aggregated as an array, data items and arrays aggregated into a GLAS Data Element, GLAS Data Elements aggregated as an EOS Data Parameter, or EOS Data Parameters aggregated into a Data Product record.
array	An ordered arrangement of homogenous data items that may either be synchronous or asynchronous. An array of data items usually implies the ability to access individual data items or members of the array by an index. An array of GLAS data items might represent the three coordinates of a georeference location, a collection of values at a rate, or a collection of values describing an altimeter waveform.
file	A collection of data stored as records and terminated by a physical or logical end-of-file (EOF) marker. The term usually applies to the collection within a storage device or storage media such as a disk file or a tape file. Loosely employed it is used to indicate a collection of GLAS data records without a standard label. For the Level 1A Data Product, the file would constitute the collection of one-second Level 1A data records generated in the SDPS working storage for a single pass.
header	A text and/or binary label or information record, record set, or block, prefacing a data record, record set, or a file. A header usually contains identifying or descriptive information, and may sometimes be embedded within a record rather than attached as a prefix.
item	Specifically, a data item. A discrete, non-decomposable unit of data, usually a single word or value in a data record, or a single value from a data array. The representation of a single GLAS data value within a data array or a GLAS Data Element.
label	The text and/or binary information records, record set, block, header, or headers prefacing a data file or linked to a data file sufficient to form a labeled data product. A standard label may imply a standard data product. A label may consist of a single header as well as multiple headers and markers depending on the defining authority.
Level 0	The level designation applied to an EOS data product that consists of raw instrument data, recorded at the original resolution, in time order, with any duplicate or redundant data packets removed.
Level 1A	The level designation applied to an EOS data product that consists of reconstructed, unprocessed Level 0 instrument data, recorded at the full resolution with time referenced data records, in time order. The data are annotated with ancillary information including radiometric and geometric calibration coefficients, and georeferencing parameter data (i.e., ephemeris data). The included, computed coefficients and parameter data have not however been applied to correct the Level 0 instrument data contents.

Level 1B	The level designation applied to an EOS data product that consists of Level 1A data that have been radiometrically corrected, processed from raw data into sensor data units, and have been geolocated according to applied georeferencing data.
Level 2	The level designation applied to an EOS data product that consists of derived geophysical data values, recorded at the same resolution, time order, and geo-reference location as the Level 1A or Level 1B data.
Level 3	The level designation applied to an EOS data product that consists of geophysical data values derived from Level 1 or Level 2 data, recorded at a temporally or spatially resampled resolution.
Level 4	The level designation applied to an EOS data product that consists of data from modeled output or resultant analysis of lower level data that are not directly derived by the GLAS instrument and supplemental sensors.
metadata	The textual information supplied as supplemental, descriptive information to a data product. It may consist of fixed or variable length records of ASCII data describing files, records, parameters, elements, items, formats, etc., that may serve as catalog, data base, keyword/value, header, or label data. This data may be parsable and searchable by some tool or utility program.
orbit	The passage of time and spacecraft travel signifying a complete journey around a celestial or terrestrial body. For GLAS and the EOS ALT-L spacecraft each orbit starts at the time when the spacecraft is on the equator traveling toward the North Pole, continues through the equator crossing as the spacecraft ground track moves toward the South Pole, and terminates when the spacecraft has reached the equator moving northward from the South Polar region.
model	A graphical representation of a system.
module	A collection of program statements with four basic attributes: input and output, function, mechanics and internal data.
parameter	Specifically, an EOS Data Parameter. This is a defining, controlling, or constraining data unit associated with a EOS science community approved algorithm. It is identified by an EOS Parameter Number and Parameter Name. An EOS Data Parameter within the GLAS Data Product is composed of one or more GLAS Data Elements
pass	A sub-segment of an orbit, it may consist of the ascending or descending portion of an orbit (e.g., a descending pass would consist of the ground track segment beginning with the northernmost point of travel through the following southernmost point of travel), or the segment above or below the equator; for GLAS the pass is identified as either the northern or southern hemisphere portion of the ground track on any orbit
PDL	Program Design Language (Pseudocode). A language tool used for module programming and specification. It is at a higher level than any existing compilable language.
process	An activity on a dataflow diagram that transforms input data flow(s) into output data flow(s).

product	Specifically, the Data Product or the EOS Data Product. This is implicitly the labeled data product or the data product as produced by software on the SDPS or SCF. A GLAS data product refers to the data file or record collection either prefaced with a product label or standard formatted data label or linked to a product label or standard formatted data label file. Loosely used, it may indicate a single pass file aggregation, or the entire set of product files contained in a data repository.
program	The smallest set of computer instructions that can be executed as a stand-alone unit
record	A specific organization or aggregate of data items. It represents the collection of EOS Data Parameters within a given time interval, such as a one-second data record. It is the first level decomposition of a product file.
Scenario	A single execution path for a process.
Standard Data Product	Specifically, a GLAS Standard Data Product. It represents an EOS ALT-L/ GLAS Data Product produced on the EOSDIS SDPS for GLAS data product generation or within the GLAS Science Computing Facility using EOS science community approved algorithms. It is routinely produced and is intended to be archived in the EOSDIS data repository for EOS user community-wide access and retrieval.
State Transition Diagram	Graphical representation of one or more scenarios.
Stub	(alias dummy module) a primitive implementation of a subordinate module, which is normally used in the top-down testing of superordinate (higher) modules.
Structured Chart	A graphical tool for depicting the partitioning of a system into modules, the hierarchy and organization of those modules, and the communication interfaces between the modules.
Structured Design	The development of a blueprint of a computer system solution to a problem, having the same components and interrelationships among the components as the original problem has.
Subroutine	A program that is called by another program
variable	Usually a reference in a computer program to a storage location, i.e., a place to contain or hold the value of a data item.

